Due date: 5 pm on Friday, January 22, 2021 for the group report which is worth 40% and your individual contributions constitute the rest 30% (in total 70% of your overall grade).

Submission: An electronic version of your group report (saved as a single .pdf format). Your Excel file should be exported as PDF and merged with the PDF of the group report as an appendix. The electronic version is to be submitted through Blackboard.

The group report and Excel file (.xlsx) are to be sent to the course coordinators via email as well.

The second piece of assessment in this coursework is an individual report outlining your contribution to the coursework. In the individual report please discuss how you have contributed to the individual questions on the brief (e.g., what you have contributed to formulating the task as a model, and what you have implemented & solved in Excel Solver), the group report writing (e.g. which parts you have written, and which tables/figures you have created). There will be a separate submission link on Blackboard for the individual report, and each student will be submitting her/his individual report individually. We will, however, cross-check individual reports from the same group to get an overall picture on who did what in a group.

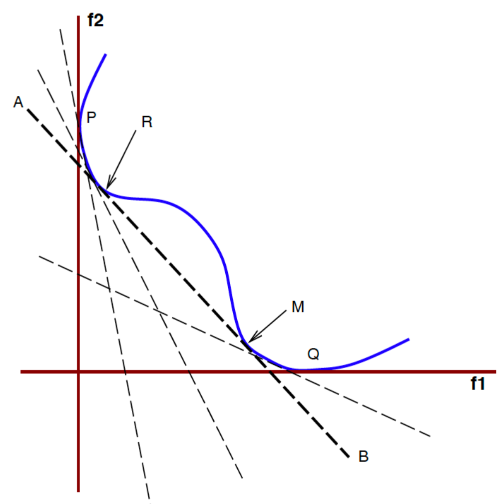
Format: Formulate a given task as an appropriate mathematical optimisation model, implement it and solve it using appropriate functionalities provided in Excel Solver. Then summarize your work using **Microsoft Word**. All core results have to be included in your report, as further detailed in the brief below. Your group report should be around **2000 words** (some bullet points will be acceptable), excluding figures, tables and references. In addition, each group member needs to write an individual report of around **500 words** describing their contribution to the project and the report. Export the Excel file as PDF, and then merge it with the PDF of the group report as an appendix. There is no limit on the length of your PDF of Excel file. For the report, use a minimum font size of Arial 11pt. Hand-written submissions will NOT be accepted.

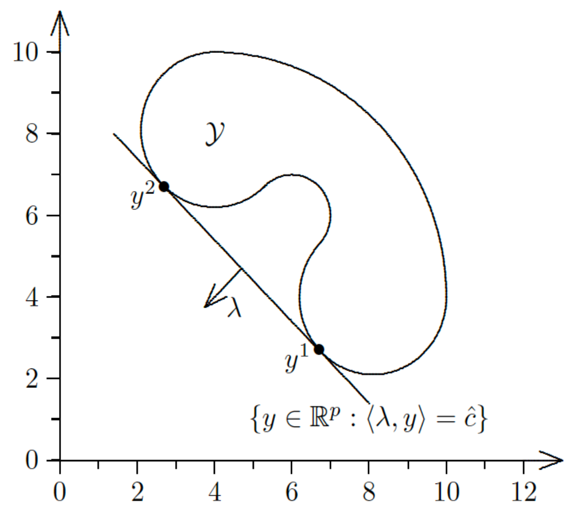
While the brief below is broken up into questions/tasks, this is done to help you in guiding your analysis and understanding what tasks must be undertaken. It does not determine the organization of your report and it should not quote the questions/tasks in the brief. Your report should be a coherent piece of work, where you describe the problem at hand, and you explain the steps taken to analyse and solve it. It should contain an introduction section and a conclusion section.

Assessment: Your coursework mark will be determined based on the quality of the group report and the individual contributions. The final coursework mark will count 70% to your overall module mark; the remaining 30% will be made up of your online tests grade. Ensure that the material presented in your report is correct and reproducible. In terms of the presentation of the report, pay attention to things such as layout, font size, keeping to the page limit, spelling, grammar, use of tables and charts, consistent number of decimal places.

Each group MUST submit their report by the deadline indicated above. Late submission will be penalised in accordance with the School's guidelines on late submission (see course description). Each group MUST work on its own. Any evidence of plagiarism will result in a mark of zero.

Questions: For any questions related to the coursework, please pose your question in **the Coursework channel** in the Teams classroom. Please do not post screenshots with pieces of code on the Teams channel as this might lead to unintentional plagiarism. Alternatively, e-mail or message on Teams the GTA: youngmin.kim@manchester.ac.uk





Brief: The aim of this coursework is to give you a glimpse into computational techniques for solving multi-objective optimization problems. Multi-objective optimization problems differ from standard, single-objective optimization problems in that it involves multiple conflicting objectives that need to be optimized at the same time. Multi-objective optimization problems typically have a set of optimal solutions. Finding these solutions can be a quite challenging task.

A commonly used approach for solving multi-objective optimization problems is the weighted sum method (WSM). This method uses a set of non-negative weights to transform a given multi-objective optimization problem into a standard, single-objective optimization problem. This process is also known as scalarization. Depending on the type of scalarized single-objective optimization problem (e.g., linear, nonlinear, mixed-integer, black-box optimization problem), appropriate algorithms can be used to solve it.

The above process of using a set of weights leads to one solution of the original multi-objective optimization. More than one solution can be found by changing the weights in a systematic way and solving the resulting single-objective optimization problems.

Formally, let us consider the following multi-objective optimization problem (which we call as MOP) with m-objectives and n variables:

$$\min f(x) = \big(f_1(x), f_2(x), \dots, f_m(x)\big),$$

$$\text{subject to } x \in X \subseteq R^n.$$

On order to find k solutions to the above MOP, we consider k different m-dimensional weights, denoted by w^j, for $j = 1,2, \dots, k$. Note that w^j is an m-dimensional real number with all components non-negative, i.e., $w^j \subseteq R^m$ and $w_i^j \geq 0$ for all $i = 1,2, \dots, m$.

Corresponding to a weight w^j, we set up the weighted sum subproblem as follows:

$$\min \sum_{i=1}^{m} w_i^j f_i(x) = w_1^j f_1(x) + w_2^j f_2(x) + \cdots + w_m^j f_m(x),$$

$$\text{subject to } x \in X \subseteq R^n.$$

The above weighted sum subproblem will be denoted by WSS^j. Let its global optimal solution be denoted as $\bar{x}^j$. It can be shown that $\bar{x}^j$ is a solution of the above MOP. Solving WSS^j for all j $(= 1,2, \dots, k)$ leads to k solutions $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^k$ of the MOP.

Description of the tasks

Real life optimisation problems in businesses often involve different aspects of problem difficulty. For example, while the classical portfolio model by Markowitz translates into a convex optimization problem with two objectives, modern portfolio optimization problems involve optimizing non-convex functions (see for example https://www.tandfonline.com/doi/full/10.1080/0013791X.2019.1636177). Another area of problem difficulty is the presence of multiple solutions that may hamper the search process of an algorithm. Depending on the type of problem difficulty, a manager needs to decide for an appropriate algorithmic approach.

The purpose of the tasks below is to show that mathematical algorithms are better for tacking some types of problem difficulty (like convexity), while metaheuristics have their niche in dealing with other types of problem difficulties. You learn about both mathematical algorithms and metaheuristics in this module. Different aspects of problem difficulties are usually modelled into a collection of test problems. There exist many such test problems, and you will be working with a well-known suite of ZDT test problems.

1. Implement test functions ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 in Excel. Descriptions of these test problems can be found at following repository: https://sop.tik.ee.ethz.ch/download/supplementary/testproblems/

2. The above test problems are scalable in the number of variables. Your experimental set-up should consider the following values: n=2, 5, 10, 30.

3. These problems involve two objectives. Therefore, you should use two-dimensional weight vectors. Based on these weight vectors, implement WSS^j (mathematically described above) in Excel.

4. Use Excel's *Nonlinear Solver* and *Evolutionary Solver* to solve WSS^j for different values of j. Use the following weight vectors:
   $w^j = \{ (0, 1), (0,25, 0.75), (0.5, 0.5), (0.75, 0.25), (1,0) \}$

5. Analyse, interpret, summarise and visualise the results. Comment on:
   a. Difficulty of finding optimal solutions for different test problems
   b. Instances where the *Nonlinear Solver* performs better and where the *Evolutionary Solver* performs better.

6. Multiple solutions to the above test problems can be found at the ETHZ repository. Explain why you are not able to obtain some solutions using the Excel's solver.

One way of visualisation is to plot the points $(f_1(\bar{x}^j), f_2(\bar{x}^j))$ for each value of j on two-dimensional plot.