

Linear Programming Applied to Humanitarian Operations Management

The objective of this project is to apply Linear Programming concepts and techniques to the planning and management of humanitarian operations in emergency situations. The project consists of developing a Linear Programming model to support decision-making on the efficient allocation and distribution of essential resources following a natural disaster.

The project should include: the mathematical formulation of the problem; the computational implementation of the model; the interpretation of the optimal solution; duality analysis; sensitivity analysis; and post-optimality analysis.

Following a major natural disaster, such as an earthquake, a large-scale wildfire, a flood, or a hurricane, an international humanitarian aid agency needs to distribute essential resources to several affected regions. The agency has several logistics centres located in different areas of the country and has limited resources available to respond to the emergency. The main resources available are, for example, drinking water; food; medicines; temporary shelter kits; hygiene supplies, etc. There are several affected regions, each with different needs and different levels of severity.

The objective is to determine how to allocate the available resources in order to minimise operational costs while ensuring minimum levels of humanitarian assistance.

Consider:

- m logistics centres, $L = \{1, 2, \dots, m\}$;
- n affected regions, $R = \{1, 2, \dots, n\}$; and
- p types of resources, $K = \{1, 2, \dots, p\}$.

Each logistic centre has a limited supply of each resource. Each affected region has minimum assistance requirements for each type of resource. Transporting resources generates costs that depend on the locations of the logistic centres and the affected regions. The objective function should minimise the total cost of the humanitarian operation.

Each group must construct its own problem instance. The instance must contain, at a minimum:

- 3 logistic centres, $m \geq 3$;
- 6 affected regions, $n \geq 6$;
- 5 types of resources, $p \geq 5$.

The groups may use: real-world data; data inspired by humanitarian operations; or data generated by the students. All data used must be properly documented and referenced.

Formulate the problem as a **linear programming** (LP) model. Define all the sets and parameters required for the formulation. Define and explain the decision variables, constraints and objective function of the model.

The model should include, at a minimum, constraints that ensure:

- the satisfaction of the minimum requirements of the affected regions;
- compliance with resource availability limits; and
- compliance with the operational capacity constraints of the logistics centres.

Create your own data set and implement the model using a computational tool of your choice.

Solve the problem using an appropriate optimisation solver (e.g., Excel Solver, PuLP in Python, Gurobi, FICO Xpress, etc.).

The solution obtained should be clearly presented and properly described. Critically analyse the solution obtained. Discuss the results from a decision-support perspective, highlighting the main insights provided by the optimisation model for the management of the humanitarian operation.

Conduct a detailed analysis of the dual solution. Interpret the shadow prices of the constraints from an economic perspective.

Perform a sensitivity analysis. Discuss the robustness of the optimal solution; critical resources; and changes in the structure of the solution.

Finally, analyse an alternative scenario that arises after the initial solution has been obtained. The scenario should represent a plausible change in the humanitarian operation and should be clearly defined and justified. For the selected scenario, present: the changes introduced to the original problem; the new optimal solution; a comparison with the original solution; and the operational implications of the changes.

Write a report of no more than eight pages. The report should be accurate, clear, and concise.

Submit both the report and the code file used to implement the model and obtain the solution.