INTERNATIONAL JOURNAL OF EXPLORING EMERGING TRENDS IN ENGINEERING
Peer-Reviewed, Refereed, Indexed and International Journal, https://ijoeete.com/
|ISSN No. 2394-0573 |Volume: 10, Issue: 2 | July - Dec 2023

CHARACTERISTIC OF MATHEMATICAL PROGRAMMING

[1] Ratnappa, [2] Rahul Dixit
[1]Research Scholar, [2]Supervisor
[1-2] Department of Mathematics, SunRise University, Alwar, Rajasthan (India)
Email: ratnappah@gmail.com

Abstract: Mathematical Programming (MP) problems involve the optimization of an objective function that depends on a set of decision variables and fixed parameters. The decision variables, that can be integer or continuous, can have bounds and/or constraints limiting the solution space. The characteristics of the variables and constraints determine the mathematical programming type.

Keywords: CHARACTERISTIC, MATHEMATICAL, PROGRAMMING

Introduction: Mathematical programming methods are based on the solution of a CAMD problem as an optimization problem where the objective function is defined in terms of the performance criteria and target properties that must be satisfied are introduced as constraints. Two solution approaches (decomposition based and direct solution based) have been mainly considered. With the decomposition-based approach, the main problem is decomposed into a set of subproblems that are solved according to a hierarchical order. Here, until the last subproblem, only solutions satisfying a subset of constraints are identified. With each solution of the subproblem, the solution space decreases. With the direct solution approach, the CAMD problems are solved using mathematical programming. Odele and Macchietto applied mathematical programming for solvent selection-design and Sinha et al. extended the MINLP models to the design of (optimal) solvent blends. Zhang et al. proposed a generic mathematical programming method for CAMD that can also be applied to solvent selection-design. It includes both first-order and second molecular groups for enhanced molecular structure representation and more accurate property estimation.
Mathematical programming is a powerful tool with broad applicability across various sectors. In this series, we will delve into the origins, relevance, and practical applications of mathematical programming, with a particular emphasis on Linear Programming (LP) and Mixed Integer Linear Programming (MIP) models.

Mathematical Programming
Mathematical Programming is a technique of mathematical optimization. Many real-world problems in such different areas as industrial production, transport, telecommunications, finance, or personnel planning may be cast into the form of a Mathematical Programming problem: a set of decision variables, constraints over these variables and an objective function to be maximized or minimized.
Mathematical Programming problems are usually classified according to the types of the decision variables, constraints, and the objective function.
A well-understood case for which efficient algorithms (Simplex, interior point) are known comprises Linear Programming (LP) problems. In this type of problem all constraints and the objective function are linear expressions of the decision variables, and the variables have continuous domains—i.e., they can take on any, usually non-negative, real values. Luckily, many application problems fit into this category. Problems with hundreds of thousands, or even millions of variables and constraints are routinely solved with commercial Mathematical Programming software like Xpress Optimizer.
Researchers and practitioners working on LP quickly found that continuous variables are insufficient to represent decisions of a discrete nature (`yes'/`no' or 1,2,3,...). This observation lead to the development of Mixed Integer Programming (MIP) where constraints and objective function are linear just as in LP and variables may have either discrete or continuous domains. To solve this type of problems, LP techniques are coupled with an enumeration (known as Branch-and-Bound) of the feasible values of the discrete variables. Such enumerative methods may lead to a computational explosion, even for relatively small problem instances, so that it is not always realistic to solve MIP problems to optimality. However, in recent years, continuously increasing computer speed and even more importantly, significant algorithmic improvements (e.g. cutting plane techniques and specialized branching schemes)

have made it possible to tackle ever larger problems, modeling ever more exactly the underlying real-world situations.

Another class of problems that is relatively well-handled are Quadratic Programming (QP) problems: these differ from LPs in that they have quadratic terms in the objective function (the constraints remain linear). The decision variables may be continuous or discrete, in the latter case we speak of Mixed Integer Quadratic Programming (MIQP) problems. In Chapters 7 and 12 of this book we show examples of both cases.

More difficult is the case of non-linear constraints or objective functions, Non-linear Programming (NLP) problems. Heuristic or approximation methods are employed to find good (locally optimal) solutions. Methods for solving non-linear problems to local optimality are Successive Linear Programming (SLP) and interior point methods. Such solvers form Xpress Nonlinear—a part of the FICO Xpress Optimization suite. If one desires to solve non-linear and mixed-integer non-linear problems to global optimality, FICO Xpress Global provides that functionality. Combined with Xpress Optimizer, FICO Xpress Nonlinear and Global can solve mixed-integer non-linear programming problems (MINLPs). However, in this book we shall not enlarge on this topic.

Building a model, solving it and then implementing the `answers' is not generally a linear process. We often make mistakes in our modeling which are usually only detected by the optimization process, where we could get answers that were patently wrong (e.g. unbounded or infeasible) or that do not accord with our intuition. If this happens we are forced to reflect further about the model and go into an iterative process of model refinement, re-solution and further analyses of the optimum solution. During this process it is quite likely that we will add extra constraints, perhaps remove constraints that we were mislead into adding, correct erroneous data or even be forced to collect new data that we had previously not considered necessary.

This paper takes the reader through all these steps: from the textual description we develop a mathematical model which is then implemented and solved. Various improvements, additions and reformulations are suggested in the following chapters, including an introduction of the available means to support the analysis of the results. The deployment of a Mathematical Programming application typically includes its embedding into other applications to turn it into a part of a company's information system.

History of Mathematical Programming

Mathematical programming, also known as mathematical optimization, originated with the invention of linear programming by George Dantzig in 1947. Since then, it has become an indispensable tool for decision-making and resource allocation in a wide range of industries, including finance, logistics, manufacturing, and transportation.

The Key Components of Mathematical Programming

The field of mathematical programming encompasses a three-step process.

Create your mathematical model. You start by translating your real-world problems mathematically, defining the questions you’re asking (decision variables), your limitations (constraints), and the goals you need to achieve (objectives).

Develop algorithms. Develop algorithms that solve these mathematical programming models. Thankfully, there are many mathematical programming “solver” solutions available, including Gurobi, that include the algorithms you’ll need.

Run the algorithms. Finally, you run your model through the solver to find the answer to your problem (i.e., answers to your questions, based on your unique objectives and constraints).

The Difference Between Mathematical Programming and Computer Programming

Mathematical programming is a problem-solving approach that uses mathematical models and algorithms to optimize decision-making processes. Computer programming, on the other hand, is about writing code to create software or systems that computers can execute. While they both involve the word “programming,” they have different focuses and objectives.

Exploring Linear Programming

Linear Programming (LP) is a widely used mathematical programming technique that involves optimizing (minimizing or maximizing) a linear objective function (your defined goals) subject to a set of linear constraints

(your defined limitations). LP is particularly useful in situations where resources need to be allocated efficiently or where decisions need to be made to maximize or minimize a certain outcome.
To illustrate the concepts of LP, we will introduce a typical case study known as the "Furniture Problem." Throughout this series, we will use the Furniture Problem to demonstrate the step-by-step process of formulating and solving LP models. By applying LP techniques to this practical scenario, you will gain a comprehensive understanding of how mathematical programming can be applied in real-world situations.
In addition to the case study, we will provide a general formulation for LP and MIP problems. Understanding the basic structure and components of LP models will enable you to tackle a wide range of optimization problems effectively.

REFERENCES

1. P.Bhattacharya, Some remarks on fuzzy graphs, Pattern Recognition Letters, 6 (1987)297-302.
2. H.Enomoto, A.S.Llado, T.Nakamigawa and G.Ringel, Super edge –magic graph, SUT Journal of Mathematics, 34(2) (1998) 105-109.
3. A.Kotzig and A.Rosa, Magic valuations of finite graph, Canad. Math. Bull., 13 (1970) 451-461.
4. J.N.Mordeson and P.S.Nair, Fuzzy Graphs and Fuzzy Hypergraphs, Physica-Verlag, Heidelberg 2000.
5. A.Nagoorgani and V.T. Chandrasekaran, Domination in fuzzy graph, Advances in Fuzzy Sets and Systems, 1(1) (2006) 17-26.
6. A.Nagoorgani and V.T. Chandrasekaran, A First Look at Fuzzy Graph Theory, Allied Publishers Pvt. Ltd, 2010.
7. A.Nagoorgani and J.Malarvizhi, Isomorphism on fuzzy graphs, World Academy of Science, Engineering and Technology, 2(4) (2008) 11-28.
8. A.Nagoorgani and D.Rajalaxmi(a) subahashini, Properties of fuzzy labeling graph, Applied Mathematical Sciences, 6 (70) (2012) 3461–3466.
9. A.A.G. Ngurah, A.N.M Salman, L.Susilowati, Super magic labelings of graphs, Discrete Mathematics, 310 (2010) 1293 – 1300.
10. A. Rosenfeld, Fuzzy Graph, In: L.A. Zadeh, K.S. Fu and M.Shimura, Editors, Fuzzy Sets and their Applications to Cognitive and Decision Process, Academic Press, New York (1975) 77-95.
11. J.Sedlacek, Theory of graphs and its applications, Smolenice Symposium (Prague, 1964),163-164, problem 27.
12. S.Avadayappan and Jayanthi P., Super magic strength of a graph, Indian Journal of Pure and Applied Mathematics, 32 (11) (2001) 1621-1630.
13. B.M.Stewart, Magic graph, Can. J.Math., 18 (1966) 1031-1059.