

Interdependence of Software and Progress of Mathematics in OR: Some Illustrative Cases and Challenges

Santosh Kumar¹, Elias Munapo²

¹Department of Mathematical and Geospatial Sciences, School of Sciences, RMIT University, Melbourne, Australia

²School of Economics and Decision Sciences, North West University, Mafikeng Campus, Mafikeng, South Africa

Email: santosh.kumarau@gmail.com, emunapo@gmail.com

How to cite this paper: Kumar, S. and Munapo, E. (2021) Interdependence of Software and Progress of Mathematics in OR: Some Illustrative Cases and Challenges. *American Journal of Operations Research*, 11, 110-119.

<https://doi.org/10.4236/ajor.2021.112007>

Received: January 28, 2021

Accepted: March 21, 2021

Published: March 24, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper points out that delayed or no supply of software can kill potential benefits associated with new mathematical ideas that have led to development of new mathematics in OR. It also points out that it is a self-created situation by the scientific community. This situation needs attention and should be resolved urgently. Many illustrative examples have been given to justify the claim.

Keywords

Linear and Integer Programs, Software Development, Protean Systems, Analysis of Protean Systems

1. Introduction

Many diverse problems that arise in various fields such as psychology, chemistry, industrial engineering, transportation planning, management, marketing, and operations research etc. can be formulated as a linear programming model or an integer programming model. Some of these models can be analyzed by the existing OR methodology, while other models create new challenges for their analysis. For example, mathematical models have given birth to ideas like Extreme Point Mathematical programming (see Kirby and Scobey [1], Kirby *et al.* [2]) and two-level resource control linear programming (see Narula and Nwosu [3]). Therefore, a demand for an improved methodology or development of new methodology remains as a challenge for the researchers in the field of OR and other fields of science and technology. In response to these new evolving situations and challenges, the field of mathematical programming and associated

analytical tools are always evolving. These changing situations and their mathematical models can be categorized as protean systems. Analysis of these ever-evolving systems requires solution approaches also to evolve with them and remain meaningful in the fast-ever-changing environment. The word “Protean” originated from “Proteus”—the legendary God of sea who is believed to have the power of assuming different shapes (see Kalyan and Kumar [4], Kumar [5], Kumar and Arora [6], Kumar and Bappoo [7], and Kumar [8] [9]). Analysis aspect of mathematical models representing complex situations is to improve the existing methods, develop new search strategies and improve search efficiency. The problems arising in business and commerce are always evolving, and to meet these new challenges, methods are also evolving in response to challenges. These complexities are making decision making process more demanding, and generally software dependence is also increasing each day. These new mathematical models create research opportunities for developing new methods and new software for dealing with large-size real-life problems and to obtain their solutions in real-time. A bad decision or a delayed decision can cause a major harm to the very existence of a company. Mathematical methods in Operations Research are developed to deal with these complexities and provide appropriate answers within a reasonable time, highlighting importance of software to deal with large-size problems arising in commerce and industry. The very nature of these methods is such that they are not only evolving in response to emerging situations, the technology and knowledge transfer [10] also brings changes to these methods. Unfortunately, software developers are either not able to keep pace with these advances in mathematical methods or these mathematical upgrades of mathematical methods go unnoticed by them or they turn a blind eye towards them. Whatever may be the reason, consequence is that many of these mathematical advances are not availed by the practitioners, thus killing valuable opportunities for real-life applications. Many mathematical ideas, which have the potential for further exploration, are not advanced further. Unfortunately, this mismatch in advancement of methods and their proper software support is causing hindrance in further refinements of the mathematical methods and stopping use of modified available methods by its potential users. Applications of these new mathematical methods can become meaningful, when they are fully supported by proper software for users to implement them on the real-life, large-size problems faced by them. We live in a protean world, where demand on modelling and analysis is ever changing, knowledge transfer can make some existing methods more versatile and may respond well to new situations. Timely software support from software developers is not coming through. It means, users are deprived of potential benefit they can have from existing methodologies. This hinderance is caused by delayed or no response from software companies to newly developed mathematical methods in OR.

Many mathematical developments go unnoticed when software support is not provided for a newly developed mathematical idea. This is a kind of self-created

injustice to development of mathematics of OR and their users in real life applications.

This paper has been organized in 6 sections. In Section 2, we outline a few reasons for the present situation and in Sections 3 and 4, we also present a few cases to illustrate our claim. In Section 5, we outline a few purposes for these new developments. These cases are from some advances in linear and integer programming methods. We believe, these mathematical ideas have high potential, which has not been realized by its users. These contributions discussed in Sections 3 and 4 have remained as theoretical contributions, as their potential benefits based on the mathematical ideas have been denied to their potential users. Finally, the paper is concluded in Section 6.

2. Reasons for Mismatch between Software Development Associated with New Mathematical Ideas

The mathematics behind mathematical methods is usually well documented, therefore researchers can make modifications to the existing methods and make these methods more versatile. However, the situation with software development is not the same. Software companies keep their coding copyrighted and sealed. They only display a limited input and output information. Mathematical developments in methods can be implemented by the software companies, who have full coding access. They can easily add the additional coding to accommodate a defined modification or refinement in a mathematical approach. If software companies do not take this step to upgrade their software within a reasonable time frame, the mathematical developments are not able to reach to their potential users. Many developments do not reach to its users as many users are not necessarily interested in the mathematical developments, but their interest lies in using new mathematical ideas if they are user-friendly. Therefore, the mathematical developments not only go unnoticed, potential subsequent progress based on that conceptual idea is also denied of future advancements based on that idea. Many researchers may argue, why depend on software companies, why not provide the full support, *i.e.*, mathematical algorithm together with its software support. This argument does not go far for two reasons:

- 1) A new coding will not stand compared to an existing coding developed by the professionals and that has been tested, modified, and eventually has become a bug-free coding.
- 2) Scientific community does not consider coding of existing methods as a significant scientific contribution.

In other words, mathematical development is like a building block, but an associated software in the present environment is not. Consequences are that many mathematical developments go unnoticed. It is a self-created situation by the scientific community. It is neither helping the users, nor is beneficial for advancing the mathematical of OR. In Sections 3 and 4 we discuss a few developments, which have not been utilized fully by their potential users.

3. Considerations of a Linear Programming (LP) Model

3.1. Formulation of a Sub-LP Model from a Given LP Model

Consider a general LP model as given by (1)

$$\begin{aligned} \text{Max } z &= \sum_{j=1}^n c_j x_j \\ \text{Subject to: } &\sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, 2, \dots, m, \quad x_j \geq 0, j = 1, 2, \dots, n \end{aligned} \quad (1)$$

It may be noted that from the LP Model (1), one can generate $n_{c_{n_1}}$ number of LP models of n_1 number of variables such that $1 \leq n_1 \leq n$. For example, one of these models can be expressed as shown by the LP Model (2).

$$\begin{aligned} \text{Max } z &= \sum_{j=1}^{n_1} c_j x_j \\ \text{Subject to: } &\sum_{j=1}^{n_1} a_{ij} x_j \leq b_i, i = 1, 2, \dots, m, \quad x_j \geq 0, j = 1, 2, \dots, n_1 \end{aligned} \quad (2)$$

All these $n_{c_{n_1}}$ number of LP problems share one common property that a feasible solution of the LP Model (2) is also a feasible solution to the LP Model (1). Suppose a feasible solution to the LP Model (2) is given by:

$$x_j = \beta_j \geq 0, j = 1, 2, \dots, n_1 \quad (3)$$

From the solution as given by (3), we can develop a new solution as given by (4):

$$x_j = \beta_j, j = 1, 2, \dots, n_1 \quad \text{and} \quad x_j = 0 \quad \text{for} \quad j = n_1 + 1, n_1 + 2, \dots, n-1, n. \quad (4)$$

The solution given by (4) is also a feasible to LP model (1). Proof is immediate as any feasible solution of the LP model (2) can be modified as a feasible solution to the LP model (1) as shown by (4). This conceptual idea was used for solving a mixed-integer programming model by solving a pure integer programming model, see Nyamugure *et al.* [11]. This simple concept of developing sub-problem from a given LP model can be applied in other situations. This is illustrated in Section 3.2 and some existing models, perceived to be restrictive, are made more versatile.

3.2. Large-Scale Linear Optimization by Transforming “n” Variable LP Problem into a 2-Variable LP Problem

Munapo and Kumar [12] considered a LP model of the form given by the LP Model (5):

$$\text{Maximize } z = CX$$

$$\text{Such that } AX \leq B$$

$$\text{where } A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}, C = [c_1 \quad c_2 \quad \cdots \quad c_n], X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (5)$$

$$a_{ij} \geq 0 \text{ for all } (i, j), i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, n;$$

$$\beta_i \geq 0, c_j \geq 0 \text{ for } i \text{ and } j, \text{ and } x_j \geq 0 \text{ for all } j.$$

They transformed the given n -dimensional LP Model (5) into a 2-variable LP problem given by (6):

$$\begin{aligned} & \text{Maximize } C \left[P_0 + \mu D + \gamma C^T \right] \\ & \text{Such that } A \left[P_0 + \mu D + \gamma C^T \right] \leq B \\ & P_0 + \mu D + \gamma C^T \geq 0 \end{aligned} \quad (6)$$

Here P_0 is a known point on one of the given constraints, D and C^T are directions of the normal of two known hyperplanes, and μ and γ are the only two unknowns to be determined, where A is the given matrix and C^T is the given column vector. Since CP_0 is a constant quantity, the Model (6) can be expressed as given by (7):

$$\begin{aligned} & \text{Maximize } \mu CD + \gamma CC^T \\ & \text{Such that } \mu AD + \gamma AC^T \leq B - AP_0 \\ & \mu D + \gamma C^T \geq -P_0 \\ & \mu, \gamma \geq 0 \end{aligned} \quad (7)$$

In this way they transformed a n -variable to a 2-variable problem. However, the non-negative requirement on all elements was considered restrictive for the real-life applications.

3.3. Denied Applications Due to Non-Availability of the Software

It may be noted that the non-negative requirement for all coefficients in (5) seems to be demanding to satisfy for all n variables in a practical situation, but the same non-negative condition may be easily satisfied for a subset of given variables. Suppose the non-negative requirements are met by a sub-set n_1 of n variables. In other words, a subset of the variables $n_1 \leq n$ may satisfy non-negative requirements of the LP Model (5). In that case, one can develop a sub-LP problem and solve that sub-problem by the defined transformation developed by Munapo and Kumar [13] and obtain some useful information concerning the given LP model. Here is a trivial illustration:

Example 1: Consider the LP Model (8)

$$\begin{aligned} & \text{Max } Z = 3x_1 + 4x_2 - x_3 - x_4 \\ & \text{Subject to : } x_1 + x_2 + x_3 + x_4 \leq 2 \\ & 2x_1 + 3x_2 + x_3 + x_4 \leq 6 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned} \quad (8)$$

One can easily verify that the optimal solution to the LP Model (8) is given by $x_1 = 2, x_2 = 0, z = 8$ and all other variables are zero. The variable x_2 is a slack variable in the 2nd constraint.

It may be noted that from LP Model (8), one can satisfy restrictions of the LP Model (5) with respect to variables x_1 and x_2 , which is given as the LP Model (9).

$$\begin{aligned}
 \text{Max } Z &= 3x_1 + 4x_2 \\
 \text{Subject to : } &x_1 + x_2 \leq 2 \\
 &2x_1 + 3x_2 \leq 6 \\
 &x_1, x_2 \geq 0
 \end{aligned} \tag{9}$$

The Model (9) can be solved by the transformation developed by Munapo and Kumar [12] and verify the optimal solution given by: $x_2 = 2, s_2 = 0, z = 8$, which is certainly a lower bound of the LP Model (8).

Munapo, *et al.* [13] further extended the approach by Munapo and Kumar [11] and developed a one-dimensional search with non-negative restriction limited to objective coefficients only. They further developed a hybrid-search of this one-dimensional search combined with simplex extreme point search. The reformulation of the subproblem from (8) will once again be the same as Model (9) and therefore the problem (9) can also be solved by the one-dimensional search process developed by Munapo *et al.* [13].

These two methods (Munapo and Kumar [12] and Munapo *et al.* [13]) have scope for real-life applications, particularly in view of the reformulation as illustrated by the LP Models (8) and (9). It raises several questions, such as:

- 1) Why the software developers have not provided the software support to these two versatile variations to LP approaches, *i.e.*, Munapo and Kumar [12] and Munapo *et al.* [13]?
- 2) Why such a useful approach has not been explored further?
- 3) Why the practitioners have been deprived of using these theoretical ideas in their real-life applications?
- 4) How are decisions made for software development in the field of OR? Why and how some ideas are deprived to flourish further to reach their justified potential?

Answer to all questions is that the scientific community needs to readdress the problem of software development or supply to researchers existing codes in some limited form. Current regulations are unsatisfactory, and they are not helping advancement of mathematical methods in OR.

4. Considerations of Linear Integer Programming Solution Approaches

In this section, we highlight a few linear integer programming solution approaches, which have been subjected to similar fate, as discussed in Section 3.

Young [14] [15] developed an interesting idea for solving a linear integer program. His approach was simple, for example, he selected the source row and column, exactly like it is done in the simplex approach. However, instead of the usual simplex pivot, Young developed a Gomory [16] constraint and selected the pivot element from the Gomory constraint, which has the co-efficient value 1. Therefore, in Young's approach, the pivot element is always 1. Consequence was that if the simplex matrix was comprised of integer elements, the new matrix after the pivoting, will remain integer elements matrix. It was a breakthrough, as

Young gave birth to an interesting concept of searching for an integer solution along the integer polyhedron. Such an important and fundamental concept has gone unnoticed, and possible reason could be that software support was not provided.

A limitation with Young [15] approach was that with each iteration, the problem dimension was increased by 1 additional constraint and 1 extra variable. This may have been a limitation, but the concept of forcing the pivot element 1 was too good to be ignored. Researchers can overcome difficulties in many ways. A good example is Gomory [16] constraint, which led to many interesting ideas discussed, see Ladanyi *et al.* [17]. Completely and independently Munapo and Kumar [18] developed a new search process for a linear integer program on extreme points of the integer polyhedron. They developed a 3-step algorithm as follows:

Step 1: Assume that all $b_i \geq 0$. Select a pivot element as is selected in the simplex approach. If the pivot is element is >1 , go to step 2, else when it is 1, go to step 3.

Step 2: Consider the pivot element $a_{ij} > 1$ and pivot row is given by:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n + s_i = b_i \quad (10)$$

The iteration is comprised of replacing the variable x_j by the variable x'_j as shown by the relation (13). Details are given below:

When the elements in the pivot row are divided by the pivot element >1 , we have:

$$\frac{a_{i1}}{a_{ij}}x_1 + \frac{a_{i2}}{a_{ij}}x_2 + \cdots + x_j + \cdots + \frac{a_{in}}{a_{ij}}x_n + \frac{1}{a_{ij}}s_i = \frac{b_i}{a_{ij}} \quad (11)$$

Separating the fractional and integer parts in each term of (11), keeping fractional parts positive, one gets the relation (12):

$$\begin{aligned} & (d_{i1} + f_1)x_1 + (d_{i2} + f_2)x_2 + \cdots + x_j + \cdots + (d_{in} + f_n)x_n + (d_i + f_i)s_i \\ & = (d + f) \end{aligned} \quad (12)$$

From (12), one gets (13):

$$x_j = d - (d_{i1}x_1 + d_{i2}x_2 + \cdots + d_{in}x_n + d_i s_i) - x'_j \quad (13)$$

Step 3: Apply the usual pivot operation as carried out in the simplex and move to Step 4.

Step 4: Check if the objective row has negative elements? If yes, an optimal integer solution has been found, else return to Step 1.

The interesting part of Munapo, Kumar and Khan [18] approach is that dimension of the model after each iteration is not increasing, as was the case in Young's approach [14] [15]. Pivoting operation is performed on 1, hence all search is free of fractions. Only some substitution relations are maintained for the final answer to be interpreted back in terms of the given variables. Once again, this idea has not been utilized for solving real-life problems raises similar concerns as were discussed in Section 3.3.

5. Research Processes and Methods

It may be noted that research process is an ongoing protean process, where real-life problems are attempted by some existing methods supported by user-friendly software. If answers are unsatisfactory, challenges are noted by the researchers, and new methods are developed. Therefore, software plays a central role, and when appropriate software is missing, the ongoing research comes to a halt and all future developments based on that idea do not move further. A typical case is that of the “Extreme Point Mathematical Programming model”, which in mathematical form is given by:

$$\text{Max } Z = CX$$

Subject to: $AX \leq b$

where X is an extreme point of

$$DX \leq f, X \geq 0. \quad (14)$$

Several methods were suggested for solve the Model (14), see Kirby and Scobey [1], Kirby *et al.* [2] and Kumar and Wagner [6]. Applications were pointed by Chandrasekhar *et al.* [19]. However, no software support was provided and in almost 4 to 5 decades the concept has not progressed much further.

6. Concluding Remarks

In this paper, we have pointed out that the OR literature, like any other field of science, evolves but many mathematical methods do not get justice to reach their potential. They are not fully utilized by their users. Many important ideas are not researched for further improvements and not used by their potential users simply due to non-availability of the software for them to experiment with the real-life problem faced by the users.

The process of software development and support needs modification and re-thinking. Non-availability of software has killed many good mathematical ideas, see instances discussed in Sections 3 and 4. Present system of software development is failing in providing appropriate software support to new and upcoming methods in OR.

Apparently, many restrictions in mathematical models may have given a wrong impression that it is not suitable for real-life applications, but this kind of thought is untrue, as variations of a model may still provide some important and useful information in manageable real-time. It is a challenge. It calls for further creativity on the part of the user. Restrictions are just challenging their users, which may call further insight to remove them or slightly modify them to obtain some useful information, in real-time. Every situation is different.

The software support for newly developed methods in the current form is unsatisfactory, and it needs attention of the scientific community now. Lots of good ideas have been wasted by paying no attention to software supply situation.

Acknowledgements

The authors are grateful to the referee for providing many useful suggestions to

improve the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Kirby, M.J.L. and Scobey, P.F. (1970) Production Scheduling of n Identical Machines. *Journal of Canadian Operational Research Society*, **8**, 14-27.
- [2] Kirby, M.J.L., Love, H.R. and Swarup, K. (1972) Extreme Point Mathematical Programming Problem. *Management Science*, **18**, 540-549.
<https://doi.org/10.1287/mnsc.18.9.540>
- [3] Narula, S.C. and Nwosu, A.D. (1991) Two Level Resource Control Pre-Emptive Hierarchical Linear Programming Problem: A Review. In: Kumar, S., Ed., *Recent Developments in Mathematical Programming*, Gordon and Breach Science Publishers, Melbourne, 29-44.
- [4] Kalyan, R. and Kumar, S. (1991) A Study of Protean Systems—Some Heuristics and Strategies for Redundancy Optimization. In: Kumar, S., Ed., *Recent Developments in Mathematical Programming*, Gordon and Breach Science Publishers, Melbourne, 181-198.
- [5] Kumar, S. (1995) Optimization of Protean Systems: A Review. *APORS'94*, Fukuoka, 26-29 July 1994, 139-146.
- [6] Kumar, S. and Arora, H. (1995) Shortest Paths and Connected Graphs in Protean Networks. In: Kapur, P.K. and Sehgal, V.K., Eds., *Operations Research—Theory and Practice*, Spaniel Publishers, New Delhi, 125-135.
- [7] Kumar, S. and Bapoo, R. (1995) Shortest Routes in Protean Networks. In: Du, D.-Z., Zhang, X.-S. and Cheng, K., Eds., *Operations Research and Its Applications*, World Publishing Corporation, Beijing, 228-237.
- [8] Kumar, S. and Wagner, D. (1979) Some Algorithms for Solving Extreme Point Mathematical Programming Problems. *New Zealand Journal of OR*, **7**, 127-149.
- [9] Kumar, S. (2005) Information Recycling Mathematical Methods for Protean Systems: A Pathway Approach. *South African Journal of Industrial Engineering*, **16**, 82-101. <https://doi.org/10.7166/16-2-168>
- [10] Treen, A. (2020) Knowledge Transfer Partnerships—An Introduction. *Mathematics Today*, **56**, 192-193.
- [11] Nyamugure, P., Munapo, E., Lesaoana, M. and Kumar, S. (2016) A Heuristic for a Mixed-Integer Program Using the Characteristic Equation Approach. *International Journal of Mathematical, Engineering and Management Sciences*, **2**, 1-16.
<https://doi.org/10.33889/IJMEMS.2017.2.1-001>
- [12] Munapo, E. and Kumar, S. (2013) Solving Large-Scale Linear Optimization Problem with Non-Negative Coefficients by Transforming into a Two-Variable LP Problem. *ASOR Bulletin*, **32**, 1-12.
- [13] Munapo, E., Kumar, S., Lesaoana, M. and Nyamugure, P. (2014) Solving Large-Scale LP Model with Non-Negative Coefficients: An Hybrid Search over the Extreme Points and the Normal Direction of the Given Objective Function. *ASOR Bulletin*, **33**, 11-23.
- [14] Young, R.D. (1968) A Simplified Primal (All Integer) Integer Programming Algo-

-
- rithm. *JORSA*, **16**, 750-782. <https://doi.org/10.1287/opre.16.4.750>
- [15] Young, R.D. (1969) Primal Integer Programming. In: Hu, T.C., Ed., *Integer Programming and Network Flows*, Addison Wesley, Boston, 287-309.
- [16] Gomory, R.E. (1958) Outline of an Algorithm for Integer Solutions to Linear Programs. *Bulletin of the American Mathematical Society*, **64**, 275-278. <https://doi.org/10.1090/S0002-9904-1958-10224-4>
- [17] Ladanyi, L., Ralph, T.K. and Trotter, L.E. (2001) Branch, Cut and Price: Sequential and Parallel. In: Naddef, N. and Juenger, M., Eds., *Computational Combinatorial Optimization*, Springer, Berlin, 223-260. https://doi.org/10.1007/3-540-45586-8_6
- [18] Munapo, E., Kumar, S. and Khan, L. (2010) Pure Integer Programming on Integer Polyhedron. *AIMS International Journal of Management*, **4**, 135-144.
- [19] Chandrasekaran, R., Kumar, S. and Wagner, D. (1980) Critical Path Problem under Assignment Constraints—An Application of an Extreme Point Mathematical Programming Problem. *Journal of Information and Optimization Sciences*, **1**, 41-51. <https://doi.org/10.1080/02522667.1980.10698659>