

Product Portfolio Management Using Linear Programming

Satyam Jain¹, Shourya², Shaurya Kalra³, Sivaan Choksi⁴, Seyan Bhushan⁵
^{1,2,3,4,5}NMIMS

I. INTRODUCTION TO OPERATIONS RESEARCH AND THE PROBLEM TYPE

1.1 Operations Research and Linear Programming

Operations Research (OR) is the use of mathematical models and analytical methods to help managers choose the best course of action when resources are limited. It grew out of the Second World War, when planners used quantitative methods to allocate scarce equipment and personnel, and was later taken up by businesses for problems such as production planning, inventory control and logistics (Hillier & Lieberman, 2015). An OR study usually follows the same pattern: define the decision, state a measurable objective, write down the limits within which the decision must be made, and use a solution technique to find the best answer that respects those limits.

Linear Programming (LP) is one of the most widely used OR techniques. An LP model has three parts: decision variables, which are the quantities the manager controls; a linear objective function, which is to be maximized or minimized; and a set of linear constraints, which represent limited resources or requirements such as demand. Dantzig's simplex method (Dantzig, 1963) made such models solvable systematically, and today software such as Excel Solver or Python solves them in seconds. When some decisions must be whole numbers or yes/no choices, the model becomes an Integer Programming (IP) or Mixed-Integer Programming (MIP) problem. These keep the LP structure and are solved by repeatedly solving LP versions of the problem.

1.2 Product Portfolio Management as a Linear Programming Problem

Product Portfolio Management (PPM) is often discussed as a strategic or marketing activity: deciding which products a company should offer and how they

fit together. Cooper, Edgett and Kleinschmidt (1999), in a large study of how businesses manage their new-product portfolios, assessed how satisfied managers were with their portfolio methods and noted that some firms faced major problems in portfolio management. Once a company has a set of possible products, however, PPM becomes a very practical operational question:

Given limited raw materials, labour hours, machine hours and market demand, which products should the company make, and how many of each, so that total contribution is as high as possible?

This is the classic product-mix problem, which appears first among the Linear Programming topics suggested in the project brief. The connection can be shown as a chain: PPM → product-mix decision → allocation of scarce resources → LP model → optimal product combination → maximum contribution and efficient use of resources. The reason LP is needed is that every product competes for the same limited resources. Making more of one product leaves less capacity for the others. A manager cannot simply rank products by profit per unit, because a product with a high unit profit may use a large amount of the scarcest resource. LP solves this by looking at all products and all constraints at the same time. In the terms used by the brief, the problem type is a resource-allocation LP with a maximization objective, which extends to integer programming when there are yes/no decisions about whether to include a product at all.

II. LITERATURE CONTEXT

Much of the debate on product-mix decisions in the 1990s was about whether LP was really necessary. Goldratt's Theory of Constraints (TOC) offered a simpler rule: find the bottleneck resource, calculate

each product's throughput (selling price minus raw material cost) per unit of bottleneck time, and produce the products with the highest ratio first. Luebbe and Finch (1992) compared the two on the same example and showed that TOC is a philosophy built around the bottleneck, while LP handles the objective and all constraints together; on their simple example both reached the same answer. The picture changes when there is more than one constrained resource. Plenert (1993) showed that the TOC rule can then give an inefficient mix, while integer linear programming finds a better one, and Linhares (2009) argued that because the product-mix decision is combinatorially complex, no simple TOC-style rule can guarantee the optimum in general.

A second line of research asks how costs should enter the model. Activity-based costing (ABC) charges labour and overhead to products, while TOC treats them as fixed. Kee and Schmidt (2000) showed that these two views can recommend opposite product mixes, and that the correct choice depends on how much control management has over its labour and overhead. For PPM these matters, because it means the profit-per-unit figures in an LP objective are a modelling choice rather than fixed facts.

A third line looks at the breadth of the portfolio rather than production quantities. Firms with thousands of products face complexity costs, such as extra inventory and longer lead times, that grow with variety. Ward et al. (2010) describe how Hewlett-Packard (HP) used OR models to decide which products to keep.

Taken together, the literature establishes three points. First, when there are several constraints, formal optimization does better than rule-of-thumb ranking. Second, the result of a product-mix model depends heavily on how profit and cost are defined. Third, real portfolio decisions often include yes/no choices about including a product, which pushes the basic LP towards integer programming.

III. PAPER 1: WARD ET AL. (2010), PRODUCT PORTFOLIO MANAGEMENT AT HP

3.1 Real-Life Problem and Domain

Ward et al. (2010) was published in *Interfaces* (now the *INFORMS Journal on Applied Analytics*), a journal that reports OR applications in real organizations. The domain is the computer hardware

industry, covering HP's printer, personal computer and enterprise server businesses.

By the mid-2000s HP's product range had become very large. The paper reports more than 2,000 laser printer stock-keeping units (SKUs), more than 15,000 server and storage SKUs, and over eight million possible configure-to-order combinations in its notebook and desktop lines. New products kept being added because each one promised some extra revenue, but the costs of that variety were largely hidden. They appeared as excess and obsolete inventory, lower forecast accuracy, expediting costs and long, unpredictable order cycle times. In 2004, HP's average order cycle time was nearly twice that of its leading competitor. The authors note that the real difficulty was not only managing the variety, but measuring its true costs and benefits.

3.2 Why OR Was Suitable

The problem had a clear objective (keep the revenue customers actually generate), a clear limit (the number of products the business can support) and a very large number of interacting choices. Each customer order usually contains several products, so removing one product can make an entire order impossible to fulfil. The value of a product therefore depends on which other products are kept, and ranking products one at a time by their own sales gives a misleading answer. With millions of combinations, only an optimization model could handle the decision.

3.3 Mathematical Model and Problem Type

HP built two tools. The first, a complexity return-on-investment calculator, screens new product proposals before launch. It estimates the extra costs that variety creates and computes:

$$\text{Complexity ROI} = (\text{Incremental margin} - \text{Variable complexity costs}) \div \text{Fixed complexity costs}$$

A proposal had to clear a hurdle rate of typically 6:1 or more. This calculator uses costing and inventory models rather than optimization, so it is not the focus here.

The second tool, Revenue Coverage Optimization (RCO), decides which existing products to keep, and it is a true optimization model:

- Decision variables: $x_p = 1$ if product p is kept in the portfolio, 0 otherwise; $y_o = 1$ if order o is fully

covered (every product it needs is kept), 0 otherwise.

- Parameters: r_o = revenue of order o ; n = maximum number of products allowed in the portfolio.
- Objective: Maximize $Z = \sum_o r_o y_o$, the total revenue of orders that can still be fully met.
- Constraints: $y_o \leq x_p$ for every order o that contains product p (an order counts only if all its products are kept); $\sum_p x_p \leq n$ (portfolio size limit); $x_p, y_o \in \{0, 1\}$.
- Assumptions: recent order history (typically the previous three months) is a good guide to future demand; an order has value only if it can be fully satisfied; and the portfolio is judged by revenue covered (the tool can also use order count or margin).

Strictly, this is a binary integer program. Its link to LP is what made it solvable. The authors moved the size limit into the objective by charging a penalty λ for each product kept, giving: Maximize $\sum_o r_o y_o - \lambda \sum_p x_p$, with the variables allowed to take any value between 0 and 1. This relaxed problem is an LP. Because each constraint $y_o \leq x_p$ links just one order and one product, the constraint matrix has a special network structure and the LP returns 0–1 answer on its own. It can be solved as a minimum-cut problem on a network with a source, one node for each product, one node for each order and a sink. Minimum-cut and maximum-flow problems are themselves special cases of LP. Solving for different values of λ produces a series of portfolios that shows how much order revenue is covered at each portfolio size.

3.4 Solution Methodology, Software and Validation

A general integer programming solver was too slow for millions of orders–product pairs. With the commercial solver CPLEX, each value of λ took several minutes, and building the full trade-off curve took many hours. HP's researchers therefore developed a new algorithm, Simultaneous Parametric Maximum Flow, which solves the network problem for all values of λ together. It was written in C++ and cut run time from hours to minutes, and in a later version to seconds. Managers used it through a web-based interface, with results shown in Excel and linked to HP's financial database.

The solution was checked in several ways. The algorithm was proved to converge, and its results were compared with CPLEX on the same formulation. Input data and outputs were reviewed with product

marketing experts. RCO was also compared with four simple ranking rules, such as ranking products by their own revenue or by unit volume, and it covered noticeably more revenue than all of them. Finally, the authors argue that the strongest evidence of validity was that actual business results matched the model's predictions.

3.5 Key Findings and Implications

The paper reports that the OR tools added more than \$500 million in profit since 2005, with an ongoing benefit of about \$180 million a year. Examples include:

- LaserJet printers: SKUs fell by about 40% between 2006 and 2009, worth roughly \$20 million in annual net profit.
- Enterprise servers: more than 3,300 products were removed from a portfolio of over 10,000.
- Personal computers: a smaller recommended range cut order cycle time by four days for notebooks and two days for desktops, and each day of improvement was estimated to save at least \$38 million a year.

The main implication for PPM is that a product's value cannot be judged in isolation. A product with modest sales of its own may be essential because many large orders depend on it, and only a model that looks at all products together can show this.

IV. PAPER 2: KEE AND SCHMIDT (2000), PRODUCT-MIX DECISIONS UNDER ABC AND TOC

4.1 Real-Life Problem and Domain

Kee and Schmidt (2000), published in the *International Journal of Production Economics*, study the product-mix decision in multi-product manufacturing. The practical problem is one many manufacturers face: two well-known management accounting approaches, TOC and ABC, often recommend different product mixes for the same factory, so which one should a manager trust? Unlike the HP paper, it is not a case study of a named company: the authors illustrate a general model with a hypothetical firm, "XYZ Inc." This limitation is discussed in Section 6, but the paper is included because it states the product-mix model formally and shows how LP-based reasoning settles a real managerial conflict.

4.2 Why OR Was Suitable

The decision involves several limited resources, different requirements for each product and demand limits. TOC and ABC are both simplified rules for this problem; writing it as a mathematical program let the authors compare them on equal terms and show when each gives the best mix.

4.3 Mathematical Model and Problem Type

- Decision variables: X_{i1} = units of product i produced; $X_{i2} = 1$ if product i is produced at all, 0 otherwise.
- Parameters: p_i = price; c_{i0} = raw material cost per unit; q_{ijk} = amount of activity j at level k used by product i (unit-level activities such as labour and machine hours, or product-level activities such as engineering drawings); Q_{jk} = capacity of each activity; c_{jk} = cost per unit of each activity; D_i = maximum demand.

Under ABC, the cost of all resources used is charged to products:

$$\text{Maximize } Z = \sum_i (p_i - c_{i0}) X_{i1} - \sum_{i,j,k} c_{jk} q_{ijk} X_{ik}$$

subject to $\sum_i q_{ijk} X_{ik} \leq Q_{jk}$ (capacity), $X_{i1} \leq D_i X_{i2}$ (demand), $X_{i1} \geq 0$, $X_{i2} \in \{0, 1\}$

Under TOC the constraints are identical, but labour and overhead are treated as a fixed period cost, so the objective becomes Maximize $Z = \sum_i (p_i - c_{i0}) X_{i1} - \sum_{j,k} c_{jk} Q_{jk}$, and only throughput affects the choice. The authors then build a general model that splits each resource's capacity into a part management can control and a part it cannot. Only the controllable resources actually used are treated as product costs. ABC and TOC turn out to be the two extreme cases of this model: all resources controllable, or none.

The model assumes that costs are linear in volume, that products do not interact in production, and that activities, capacities and demand are known with certainty. The problem type is a mixed-integer linear program. Its continuous part is a standard LP product-mix model; the binary variables only decide whether a product enters the portfolio. Once those are fixed, what remains is an ordinary LP.

4.4 Solution Methodology, Software and Validation

The paper does not name any software. For the example, the authors graph the feasible region and show that labour is the binding constraint. The TOC mix is found by ranking products by throughput per labour hour, the ABC mix by comparing the corner

points of the labour constraint, and the general model is solved for the case where half of the machine hours are discretionary. The results are checked through income statements for each mix. The authors also vary the discretionary machine hours from 0 to 600,000, effectively a sensitivity analysis, and show that the general model's income is never lower than that of TOC or ABC.

4.5 Key Findings and Implications

Table 1 summarizes the example and the results reported by Kee and Schmidt (2000).

Table 1: Kee and Schmidt's (2000) XYZ Inc. example and results (hypothetical data from the paper)

Item	Product A	Product B
Price / raw material cost per unit	\$25 / \$5	\$46 / \$10
Labour hours per unit (capacity 240,000 h)	0.5	1.0
Machine hours per unit (capacity 600,000 h)	1.0	0.5
Maximum demand (units)	500,000	300,000
Throughput per labour hour	\$40	\$36

Approach	Product mix selected	Net income ^a
TOC	480,000 A	\$3,330,000
ABC	240,000 B	\$3,580,000
General model	240,000 A + 120,000 B	\$4,050,000

^aWhen half of the machine hours are a fixed commitment the firm pays for regardless of the mix.

TOC favours Product A because it earns \$40 of throughput per labour hour against \$36 for B. ABC favours B because, once labour and machine costs are charged, B earns \$18 per unit against \$8 for A. When half of the machine hours are a fixed commitment, neither answer is best: the general model makes both products and earns \$4,050,000, which is \$470,000 more than ABC and \$720,000 more than TOC. The lesson for PPM is that the profit per unit in a product-mix LP is not a fixed number. It depends on which costs really change when the mix changes. If the

objective uses the wrong costs, the model will optimize the wrong thing, however good the solution method.

V. EXPLAINING THE LP MODEL: AN ILLUSTRATIVE WORKED EXAMPLE

5.1 Problem Description and Data

To show how the basic LP product-mix model works step by step, this section uses a small example. The

numerical dataset below is an illustrative dataset created for demonstrating the OR model and is not claimed to be actual company data. It describes a hypothetical furniture manufacturer, referred to as Firm W, that makes chairs, tables and bookshelves. All three products share three scarce resources: seasoned wood, skilled labour and time on one CNC machine. The firm is planning one month's production.

Table 2: Illustrative monthly data for Firm W (hypothetical)

Product	Contribution per unit	Wood (kg)	Labour (hrs)	Machine (hrs)	Max demand
Chair (x_1)	₹900	6	2	1	300
Table (x_2)	₹2,500	20	5	3	120
Bookshelf (x_3)	₹1,800	15	3	2.5	150
Available		3,000	1,000	600	

5.2 Formulation

Decision variables: x_1 , x_2 and x_3 are the numbers of chairs, tables and bookshelves made in the month.

Objective function: contribution per unit is the selling price minus variable costs (wood, piece-rate labour, power and consumables). Fixed costs such as rent and salaries do not change with the mix in the short run, so maximizing total contribution also maximizes profit:

$$\text{Maximize } Z = 900x_1 + 2,500x_2 + 1,800x_3$$

Constraints:

$$6x_1 + 20x_2 + 15x_3 \leq 3,000 \quad (\text{wood, kg})$$

$$2x_1 + 5x_2 + 3x_3 \leq 1,000 \quad (\text{labour, hours})$$

$$x_1 + 3x_2 + 2.5x_3 \leq 600 \quad (\text{machine, hours})$$

$$x_1 \leq 300, \quad x_2 \leq 120, \quad x_3 \leq 150 \quad (\text{demand})$$

$$x_1, x_2, x_3 \geq 0 \quad (\text{non-negativity})$$

The first three constraints say that the firm cannot use more wood, labour or machine time than it has. The demand constraints stop the model from recommending production that cannot be sold, and non-negativity rules out negative output.

Assumptions: contribution and resource use per unit are constant (proportionality); total contribution and

resource use are the sums across products (additivity); fractional units are allowed (divisibility); all data are known with certainty; and the plan covers a single period.

5.3 Solution Method

With three variables the graphical method cannot be used, so the model is solved with the simplex method. The optimum of an LP always lies at a corner point of the feasible region. Simplex starts at one corner and moves to a neighbouring corner that improves Z , stopping when no neighbour is better. Here the model was solved with the HiGHS solver in Python's SciPy library (code in Appendix A); the same model can be solved in Excel Solver with the Simplex LP engine (steps in Appendix B). Along with the optimal quantities, the solver reports how much of each resource is used and the shadow price of each constraint, which is the increase in Z from one more unit of that resource.

5.4 Results and Interpretation

Table 3: Optimal solution for Firm W ($Z = ₹4,20,000$)

Constraint	Used	Available	Status	Shadow price
Wood (kg)	3,000	3,000	Binding	₹125 per kg
Labour (hrs)	900	1,000	100 spares	₹0

Constraint	Used	Available	Status	Shadow price
Machine (hrs)	480	600	120 spares	₹0
Chair demand	300	300	Binding	₹150 per unit
Table demand	60	120	60 unsold	₹0
Bookshelf demand	0	150	Not produced	₹0

The optimal plan is to make 300 chairs, 60 tables and no bookshelves, for a total contribution of ₹4,20,000. Wood is the only binding resource; labour and machine time both have spare capacity.

The interesting part is why. Tables have the highest contribution per unit (₹2,500), yet only half of their demand is met. Bookshelves earn twice as much per unit as chairs (₹1,800 against ₹900), yet none are made. The explanation lies in contribution per kilogram of wood, the scarce resource: chairs earn ₹150 per kg, tables ₹125 and bookshelves ₹120. The model therefore gives wood to chairs first until their demand is used up (1,800 kg), gives the remaining 1,200 kg to tables (60 units), and has nothing left for bookshelves. The model does not simply identify the

most profitable product. It identifies the combination of products that earns the most in total, given which resource is actually scarce.

The shadow prices also guide management. An extra kilogram of wood is worth ₹125, so buying more wood is worthwhile up to that premium over its normal cost. Extra labour or machine time is worth nothing at present, so overtime or a second machine would add cost without adding contribution. Each extra chair that marketing can sell is worth ₹150.

5.5 Sensitivity Analysis

Portfolio decisions are not one-off decisions: Table 4 re-solves the model with one change at a time.

Table 4: Sensitivity scenarios (chairs, tables, bookshelves)

Scenario	New optimal mix	Z (₹)	What happens
Base case	300, 60, 0	4,20,000	Wood is the bottleneck
1. Wood +10% (3,300 kg)	300, 75, 0	4,57,500	Gain = 300 kg × ₹125
2. Machine +20% (720 hrs)	300, 60, 0	4,20,000	No gain; machine was not scarce
3. Chair demand rises to 400	400, 30, 0	4,35,000	Wood shifts from tables to chairs
4. Bookshelf contribution ₹2,000	300, 0, 80	4,30,000	Bookshelves replace tables
5. Labour -30% (700 hrs)	125, 0, 150	3,82,500	Mix changes completely

Scenarios 1 and 3 confirm the shadow prices: the increases in Z (₹37,500 and ₹15,000) are exactly the extra units multiplied by ₹125 and ₹150. Scenario 2 shows that investing in a non-binding resource gives nothing. Scenario 4 shows that a product can move in or out of the portfolio with a small price change: once a bookshelf earns more than ₹1,875, its contribution per kg of wood exceeds that of a table, and bookshelves replace tables. Scenario 5 is the most

striking. When labour becomes scarce as well as wood, the ranking changes because bookshelves earn the most per labour hour (₹600, against ₹500 for tables and ₹450 for chairs). The firm should now drop tables and make bookshelves, the product the base model ignored. A portfolio that is optimal this month may be the wrong one next month.

VI. CRITICAL REFLECTION

Table 5: Comparison of the models studied

Feature	Ward et al. (2010)	Kee & Schmidt (2000)	Worked example
Setting	Real company (HP)	Hypothetical firm	Hypothetical firm
Decision	Which products to keep	Which products and how many	How many of each

Feature	Ward et al. (2010)	Kee & Schmidt (2000)	Worked example
Objective	Max. revenue of fully covered orders	Max. profit	Max. contribution
Problem type	Binary IP solved through its LP / min-cut form	Mixed-integer LP	LP
Method and tools	Parametric max-flow in C++; CPLEX check	Graphical / corner points; no software named	Simplex (SciPy, HiGHS)
Validation	Proofs, CPLEX, experts, business results	Income statements, parametric analysis	Resource check, sensitivity

6.1 Strengths

- Objective, transparent decisions.

The trade-offs between products are stated explicitly instead of being left to intuition or departmental pressure. At HP, this is exactly what had been missing: new products were added for their revenue while their costs stayed hidden.

- Handles many constraints at once.

In both papers, optimization beat simple ranking rules: TOC ranking in Kee and Schmidt (2000), and revenue or volume ranking in Ward et al. (2010). The worked example shows why, as the best product changes depending on which resource is scarce.

- More than one answer.

Shadow prices and sensitivity analysis tell managers where extra capacity is worth paying for and where it is not.

- Scales to real problems.

By exploiting the LP structure of its model, HP could analyze millions of orders–product pairs in minutes, and the reported impact of over \$500 million shows what this can be worth.

6.2 Limitations and Assumptions

- Linearity. LP assumes constant contribution and resource use per unit. In practice there are volume discounts and economies of scale, and HP itself found that costs per unit rise as volumes fall. For PPM these matters, because the cost of keeping a low-volume product is exactly what is in question.
- The profit figure is a modelling choice. As Kee and Schmidt (2000) show, classifying costs wrongly produces the wrong mix even when the solution is mathematically optimal.

- Certainty and a single period. Demand and capacities are treated as known. HP's model relies on the past three months of orders, so it cannot see planned launches or products nearing the end of their life, a limitation the authors themselves point out.
- Strategic and qualitative factors. Brand positioning, customer loyalty, product life cycle and cannibalization are mostly outside the model. HP estimated cannibalization subjectively through its marketing teams. In the worked example, dropping bookshelves may lose customers who buy matching sets, which the model cannot capture because it treats each product's demand as independent.
- Evidence base. Kee and Schmidt's conclusions rest on a numerical example, which the authors accept limits how far they can be generalized. HP's results are reported by the company's own team, so they should be read with some caution. HP's method also skips some portfolio sizes (it covers about 85% of possible sizes), and each business unit needed a customized version.

6.3 Possibilities for Extension

- Integer programming to require whole units and to add a fixed cost for including a product, as Kee and Schmidt's binary variables already do.
- Goal programming or multi-objective optimization to balance profit against market share, minimum output for strategically important products, or capacity utilization.
- Stochastic programming or scenario analysis to handle uncertain demand, raw material prices and supply disruptions.
- Multi-period LP to cover inventory carried between months, seasonal demand, inflation and different stages of the product life cycle.

- Linking constraints for products sold together (for example, at least one bookshelf for every five tables) and environmental constraints such as a limit on wood use or carbon emissions.

These extensions point towards a two-stage approach that combines both papers: first decide which products belong in the portfolio, as HP did, and then decide how much of each to produce under capacity limits, as in Kee and Schmidt (2000) and the worked example. In this way the basic model, maximize profit, grows into one that balances profit, growth and strategic value while allowing for uncertainty.

VII. CONCLUSION

This report treated Product Portfolio Management as an Operations Research problem: once a firm must decide which products to make and how many of each with limited resources, PPM becomes a resource-allocation problem that Linear Programming is designed to solve. The two papers show this at different levels. Ward et al. (2010) used an LP-based network model at HP to decide which products to keep, and reported more than \$500 million in added profit.

Kee and Schmidt (2000) used a mixed-integer LP to show that TOC and ABC are special cases of a more general product-mix model, and that the right mix depends on which costs management can actually control.

The worked example showed the core idea in a form that can be checked by hand: the best mix is driven by contribution per unit of the scarce resource, not contribution per unit of product, and it can change completely when conditions change.

For managers, the model offers an objective basis for portfolio decisions, along with shadow prices that show where extra capacity is worth buying. Its limits are equally clear: linear costs, certain data, a single period and the absence of strategic factors such as brand, cannibalization and product life cycle. LP is therefore best seen as the quantitative core of PPM rather than a complete answer, and integer, multi-objective, stochastic and multi-period models offer a natural path to bring it closer to real portfolio decisions.

REFERENCES

- [1] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt, "New product portfolio management: Practices and performance," *J. Product Innovation Management*, vol. 16, no. 4, pp. 333–351, 1999, doi: 10.1111/1540-5885.1640333.
- [2] G. B. Dantzig, *Linear Programming and Extensions*. Princeton, NJ, USA: Princeton University Press, 1963.
- [3] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 10th ed. McGraw-Hill Education, 2015.
- [4] R. Kee and C. Schmidt, "A comparative analysis of utilizing activity-based costing and the theory of constraints for making product-mix decisions," *Int. J. Production Economics*, vol. 63, no. 1, pp. 1–17, 2000, doi: 10.1016/S0925-5273(99)00005-5.
- [5] A. Linhares, "Theory of constraints and the combinatorial complexity of the product-mix decision," *Int. J. Production Economics*, vol. 121, no. 1, pp. 121–129, 2009.
- [6] R. Luebbe and B. Finch, "Theory of constraints and linear programming: A comparison," *Int. J. Production Research*, vol. 30, no. 6, pp. 1471–1478, 1992, doi: 10.1080/00207549208942967.
- [7] G. Plenert, "Optimizing theory of constraints when multiple constrained resources exist," *European J. Operational Research*, vol. 70, no. 1, pp. 126–133, 1993.
- [8] J. Ward, B. Zhang, S. Jain, C. Fry, T. Olavson, H. Mishal, J. Amaral, D. Beyer, A. Brecht, B. Cargille, R. Chadinha, K. Chou, G. DeNyse, Q. Feng, C. Padovani, S. Raj, K. Sunderbruch, R. Tarjan, K. Venkatraman, and J. Zhou, "HP transforms product portfolio management with operations research," *Interfaces*, vol. 40, no. 1, pp. 17–32, 2010, doi: 10.1287/inte.1090.0476.

APPENDIX A:

Python Code Used to Solve the Worked Example
The model in Section 5 was solved with the following script (Python 3 with SciPy, which uses the HiGHS LP solver). Changing the numbers in the lists reproduces each scenario in Table 4.

```
from scipy.optimize import linprog
# Contribution per unit (Rs): chair, table, bookshelf
profit = [900, 2500, 1800]
```

Resource use per unit: rows = wood (kg), labour (hrs), machine (hrs)

A = [[6, 20, 15],

[2, 5, 3],

[1, 3, 2.5]]

available = [3000, 1000, 600]

Maximum demand gives an upper bound for each product

bounds = [(0, 300), (0, 120), (0, 150)]

linprog minimizes, so the objective is multiplied by -1

```
res = linprog([-p for p in profit], A_ub=A,
b_ub=available,
bounds=bounds, method="highs")
```

```
print ("Optimal mix (chairs, tables, bookshelves):",
res.x)
```

```
print ("Total contribution Z:", -res.fun)
```

```
print ("Shadow prices (wood, labour, machine):",
```

```
[-m for m in res.ineqlin.marginals])
```

```
print ("Value of extra demand (chair, table, shelf):",
```

```
[-m for m in res.upper.marginals])
```

APPENDIX B:

Solving the Worked Example in Excel Solver

1. Enter the contribution per unit for the three products in one row (for example B2:D2) and leave three cells for the decision variables (B3:D3).
2. Enter the resource requirements per unit for wood, labour and machine in rows below (B5:D7), and the available amounts in F5:F7.
3. In E5, enter =SUMPRODUCT (B5:D5, \$B\$3:\$D\$3) and copy it down to E7. In E2, enter =SUMPRODUCT (B2:D2, B3:D3) for total contribution.
4. Open Data → Solver. Set Objective: E2, to: Max, By Changing Variable Cells: B3:D3.
5. Add constraints: E5:E7 <= F5:F7, and B3:D3 <= B9:D9. Tick "Make Unconstrained Variables Non-Negative".
6. Select the solving method "Simplex LP" and click Solve. Choose "Keep Solver Solution" and tick the Answer and Sensitivity reports to see binding constraints and shadow prices.

	A	B	C	D	E	F	G
1		Chair	Table	Bookshelf	Total Z (£)		
2	Contribution per unit (£)	900	2,500	1800	4,20,000		
3	Units to produce (Solver changes these)	100	60	0			
4	Resource	Chair	Table	Bookshelf	Used	Available	
5	Wood (kg)	6	20	15	3000	3,000	
6	Labour (hrs)	2	5	3	900	1,000	
7	Machine (hrs)	1	3	2.5	480	600	
8							
9	Maximum demand (units)	300	120	150			
10							
11	Blue cells are inputs (hypothetical data from Section 3 of the report). Yellow cells B3:D3 are the decision variables Solver changes. E2 is the objective.						
12	Expected result after Solver: B3 = 300, C3 = 60, D3 = 0, E2 = 4,20,000; wood fully used (3,000 kg).						
13							

A

B

C

D

E

F

G

H

1

Microsoft Excel 16.113 Answer Report

2

Worksheet: [Firm_W_Solver.xlsx]Model

3

Report Created: 05/10/26 9:57:28 PM

4

Result: Solver found a solution. All constraints and optimality conditions are satisfied.

5

Solver Engine

6

Engine: Simplex LP

7

Solution Time: 34.332 Seconds.

8

Iterations: 5 Subproblems: 0

9

Solver Options

10

Max Time Unlimited, Iterations Unlimited, Precision 1E-06, Use Automatic Scaling

11

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

12

13

14

Objective Cell (Max)

15

CellNameOriginal ValueFinal Value

16

\$E\$2Contribution per unit (£) Total Z (£)04,20,000

17

18

Variable Cells

19

CellNameOriginal ValueFinal ValueInteger

20

\$B\$3Units to produce (Solver changes these) Chair0300Contin

21

\$C\$3Units to produce (Solver changes these) Table060Contin

22

\$D\$3Units to produce (Solver changes these) Bookshelf00Contin

23

24

25

26

Constraints

27

CellNameCell ValueFormulaStatusIndex

28

\$E\$5Wood (kg) Used3000\$E\$5<=\$F\$5Binding0

29

\$E\$6Labour (hrs) Used900\$E\$6<=\$F\$6Not Binding100

30

\$E\$7Machine (hrs) Used480\$E\$7<=\$F\$7Not Binding120

31

\$B\$3Units to produce (Solver changes these) Chair300\$B\$3<=\$D\$9Binding0

32

\$C\$3Units to produce (Solver changes these) Table60\$C\$3<=\$D\$9Not Binding60

33

\$D\$3Units to produce (Solver changes these) Bookshelf0\$D\$3<=\$D\$9Not Binding150

34

	A	B	C	D	E	F	G	H	I
1	Microsoft Excel 16.113 Sensitivity Report								
2	Worksheet: [Firm_W_Solver.xlsx]Model								
3	Report Created: 05/10/26 9:57:29 PM								
4									
5									
6	Variable Cells								
7									
8	Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease		
9	\$B\$3	Units to produce (Solver changes these) Chair	300	150	900	1E+30	150		
10	\$C\$3	Units to produce (Solver changes these) Table	60	0	2500	500	100		
11	\$D\$3	Units to produce (Solver changes these) Bookshelf	0	-75	1800	75	1E+30		
12									
13	Constraints								
14									
15	Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease		
16	\$E\$5	Wood (kg) Used	3000	125	3000	400	1200		
17	\$E\$6	Labour (hrs) Used	900	0	1000	1E+30	100		
18	\$E\$7	Machine (hrs) Used	480	0	600	1E+30	120		
19									
20									