A NOVEL NUMERICAL PYTHON-BASED OPTIMIZED ALGORITHM FOR ADVANCING IN AGGREGATE PRODUCTION PLANNING

Rubina Feroz[1], Sana Shabbir[2], Aamir Shahzad[*3], Sidra Ashraf[4]

[1,2]Faculty of Sciences, Superior University Lahore, Lahore 54000, Pakistan
*[*3]Department of Mathematics, Faculty of Natural Science and Technology, Baba Guru Nanak University, Nankana Sahib 39100, Pakistan*
[4]Department of Mathematics, Faculty of Sciences, University of Sargodha, Sargodha 40100, Pakistan

[3]robinaferoz5@gmail.com, [2]sanamaliksgd@gmail.com, [*3]aamir.shahzad@bgnu.edu.pk, [4]sideeera@gmail.com

DOI: https://doi.org/10.5281/zenodo.18322333

Keywords
Optimization Productivity, Variable Workforce, Python Pyomo, Python SciPy, Aggregate Production Planning.

Article History
Received: 22 November 2025
Accepted: 03 January 2026
Published: 17 January 2026

Copyright @Author
Corresponding Author: *
Aamir Shahzad

Abstract
Aggregate Production Planning (APP) is the procedure that seeks to establish optimal levels of production, inventory and labor over an intermediate period in order to satisfy demand at the lowest possible cost. This study presents an optimization of aggregate production scheduling using a variable labor model. The model incorporates crucial production scheduling decisions, including work allocation, inventory management, backorders, and personnel adjustments such as hiring and firing. To solve the model, two Python-based solvers are used: SciPy and Pyomo, which apply linear programming relaxation and mixed integer programming, respectively. The study evaluates both solvers in terms of computational efficiency and solution accuracy. The results show that SciPy offers a fast approximation, while Pyomo provides accurate solutions at a higher computational cost. This comparison highlights the swap between accuracy and speed and supports the effective use of open-source solvers in real-world production scheduling framework.

INTRODUCTION

Optimization is a quantitative technique which is applied to find exceptional optimal so- lution to a problem by satisfying the given constraints. It plays a vital role in decision-making in different industries like finance, logistics and manufacturing. Aggregate production planning (APP) in production management based on optimization techniques that are use to balance demand and supply, by checking cost-effective resource allocation and effective scheduling. Linear programming(LP) is one of the most globally used optimization method for solving APP problems. It allows industries to find the optimal production plan for maximizing profit, minimizing cost and execution time by using different constraints such as raw material, labour cost, demand and supply fluctuations [1,3]. Python, a high level programming language offers many optimization solvers for linear programming problems. PuLP, SciPy and CVPXY are generally used to LP problems. By integrating it with linear discriminant analysis, Gangkou et al. (2003) [2] proposed a multiple criteria linear

programming (MCLP) approach to data mining. They developed data mining algorithms by using MCLP, described how to use them on SAS and Linux, and evaluate their performance on experimental and commercial databases.
The research demonstrated that MCLP methods can outperform induction decision tree techniques in some cases and effectively discover knowledge patterns. Stefan Vo and David L. Woodruff (2006) [3] focused on the significance of integrated and optimized decision-making within supply chain management with an emphasis on tactical production planning. The book started with simple but effective models before increasingly addressing more advanced topics such as congestion and uncertainty. It pointed out how the application of advanced optimization software for production planning had risen with improvements in computing power and information technology. At the intermediate level of a hierarchical decision-making process, the aggregate production planning difficulties were studied by Oscar S. Silva Filho et al. (2010) [4]. In order to predict managerial methods for future resource allocation, these difficulties were utilized. An Excel spreadsheet-based managerial decision support system was implemented in the study to address issues with aggregate production planning. The application of the tool was demonstrated using a case study from the literature, and its computational aspects were introduced.
Gilvan C. Souza (2014) [5] investigated supply chain management's use of advanced analytics techniques. Based on the domains of the SCOR model plan, source, make, deliver, and return the study divided these applications into three categories: descriptive, predictive, and prescriptive analytics. Real-time supply chain insights were obtained by descriptive analytics using data from GPS, RFID chips, and visualization tools. Demand forecasting at the strategic, tactical, and operational levels was the main emphasis of predictive analytics in order to assist planning procedures. By combining descriptive and predictive models, prescriptive analytics used mathematical optimization and simulation techniques to improve decision-making. Gholamian et al. (2015) [6] developed a fuzzy multi-objective mixed-integer nonlinear programming (FMOMINLP) model for supply chain production planning in the real world. The model of their's deals with uncertainty in aggregate production planning involved multi-site, multi-period, and multi-product by considering multiple suppliers, manufacturers, and customers. Four conflicting objectives cost minimization, customer satisfaction, workforce stability, and purchasing value are optimized in this research. A comparative evaluation and verification by way of a genuine industrial SC case study was made after the model was re-formulated as a multi-objective mixed-integer linear programming (MOMILP) and optimized using two methodologies.
A mathematical model for aggregate production planning in a pump manufacturing company was created by Anand Jayakumar et al. (2017) [7]. To help planners choose industrial processes and inventory strategies, a formulation based on process selection and lot-sizing models was suggested. A case study was carried out using a mixed method that permitted adjustments to labor and inventory levels during the course of the planning horizon. Python was utilized to effectively tackle the problem, and the study showed that optimization models could identify the best mixed strategy. Three methods of decision-making for inventory management, sales, and production planning were introduced by Jorge Luiz Biazzi (2018) [8]. The analysis considered a situation with limited storage space, non-stationary probabilistic demand, pro- duction capacity limitations, and revenue losses due to shortages. While the second method included safety inventories without accounting for stockouts, the first method approached the problem in a linear and predictable manner. The third method calculated the penalties re-sulting from an unpredictable demand. The study demonstrated that using Microsoft Excel solvers to create more complicated models did not always yield significant benefits. Demirel et

al. (2018) [9] investigated the effect of demand uncertainty on production planning and recommended including the Profile des exigences de flexibility (PRF) into conventional aggregated planning. They structured the issue as a linear program with a combination of whole numbers and other constraints in order to set flexible limits on production plans. The study used digital experiences from the automotive and textile industries to compare global planning based on PRF to traditional methods. Based on the results, the integration of PRF has made it possible to obtain more coherent production schedules without significantly increasing costs. Ali Cheraghalikhania et al. (2018) [10] have thoroughly examined the PPG production global- ization optimization models. Their study's goal was to pinpoint the gaps in the literature and rigorously categorize APP models. The report has illustrated the evolution of APP research since Nam and Logendran's 1992 study by highlighting modeling structures, significant issues, and solution approaches. Compared to earlier, primarily methodological studies, this one has provided a more thorough assessment of the features of APP.

Additionally, the authors have provided a roadmap for future research aimed at improv- ing the models and procedures of production aggregation (PPA) planning. Jamalnia and Soukhakian (2018) [11] developed a hybrid non- linear multi-objective floue programming model (H-FMONLP) for the planning of aggregated production (PPA) in a flou environment. Their methodology aimed to minimize costs associated with production, stocking, suffering orders, and changes in main-d'oeuvre while also increasing customer pleasure. Through an interactive decision-making process, the study has integrated the limits of stocks, demand, work- related productivity, machine capacity, and storage space. A real-world case study was used to demon- strate the model's applicability, and the GENOCOP III method was employed to solve the specific non-linear programming final problem. Parganiha K. (2018) [12] has investigated how linear programming can be used to optimize resource allocation across several sectors. The main goal of the study was to solve a problem of product range optimization using Python and the PuLP module. The goal was to find the optimal combination to maximize profit by devel- oping a mathematical model and using a solver. The best solution was located at the location where the profit was highest thanks to the use of Python to create a graphic representation of the feasible region. Sodero, Jin, and Barratt (2019) [13] have examined the social acceptance process of Big Data and Predictive Analysis (BDPA) in the management of the supply chain for detailed commerce. According to their research, the BDPA technology needed to be adjusted to organizational and social contexts before it could be used. They observed that institutional and social factors influenced the BDPA's implementation, making it challenging to transfer between cooperative organizations. The study has made it abundantly evident that managers must pay close attention to the institutional context in order to ensure effective use of the BDPA.Shete et al. (2019) [14] conducted a theoretical study on manufacturing decisions in supply chain management, emphasizing the significance of making clear decisions about stock and fabrication management. Two artificial intelligence technologies, namely retropropagating neural networks and multi couches perceptrons, have been employed in the study to develop a prediction method. Schmid et al. (2020) [15] have demonstrated the industrial potential of the open-source Python Pyomo module by modeling and comparing non-linear control strategies for the Williams-Otto process. Its work demonstrates Pyomo's ability to do dynamic simulations and optimal control, enabling the quick prototyping of process systems in an industrial numerical environment. Using real data from a Pune-based industrial product manufacturing company, the effectiveness of the proposed method in forecasting demand was illustrated. Rehman et al. (2021) [16] studied how productivity loss affects conventional aggregate pro- duction planning, which normally bases labor, production, and inventory decisions on demand forecasts without considering productivity loss. To address this issue in various sectors, the study incorporated productivity loss into aggregate production planning (APP), applied the executory evaluation approach, and linear programming (LP) methods. A fixed workforce and a flexible workforce were the two model variations discussed. To assess the impact of productivity loss, the PuLP library was used to solve mathematical models. Computational results showed that hiring and termination decisions were directly affected by production loss. To find the optimal solutions for inventory systems while respecting the imposed restrictions, Alridha et al. (2022) [17] studied optimization strategies. By using numerical optimization techniques for linear programming, the study sought to maximize net profit in production planning, particularly in the manufacturing of vegetable oil. Numerous methods were applied, such as objective functions, dense and sparse matrices, and equations. The GEKKO package was used to implement the optimization in Python. Strong convergence in profit and production line calculations was shown by the numerical findings, demonstrating the effectiveness and precision of the methods used to arrive at the best answers. With a focus on Kenya Power and Lighting Company, Patrick et al. (2022) [18] examined the use of predictive analytics in supply chain management within utility firms. Although the amount of data has increased due to information technology, they pointed out that predictive analytics's efficient use in supply chain decision-making has been hampered by the absence of a clear causal relationship. The goal of the project was to improve supply chain management decision-making by using a predictive analytics framework. The researchers created an integrated framework for big data analytics, evaluated supply chain management systems in Kenyan power businesses, and eval- uated current predictive analytics solutions. The study was carried out in the Western Region of Kenya Power and Lighting Company using a Design Science research methodology. Esteso et al. (2023) [19] proposed a theoretical structure to help mathematical programming models and academic optimization tools transition to environments that are compatible with current Industry 4.0. Their work shows how high-level tools like Pyomo can be used to effectively implement production planning models. The application of predictive analytics to improve inventory control and production scheduling in the US manufacturing sector was investigated by Bashar et al. in(2024) [20].Businesses may improve their demand forecasts, optimize their production schedules, and effectively adjust their effects by combining historical data with real-time time. This study examines the worldwide production planning of the textile industry using a variable main-of-work model that takes licenses and embauches into account. With the help of SciPy (LP relaxation) and Pyomo (MIP), a linear programming formulation is developed and solved, allowing us to evaluate solver performances and demonstrate their potential to support data-driven decision-making.

1. Problem Description and Mathematical Framework

Aggregate production planning (APP) plays a critical role in balancing production costs, workforce levels, and demand fulfillment. When workforce levels are variable, the planning process becomes more complex, requiring efficient optimization to manage multiple decision variables. Mathematical modeling tools such as Pyomo and SciPy provide different approaches to solving these problems, using mixed-integer programming (MIP) and linear programming (LP) relaxation techniques. However, there is limited research directly comparing their performance for variable workforce APP models. This study formulates the problem in both MIP and LP relaxation forms and evaluates the two solvers in terms of solution accuracy, computation time, and applicability, aiming to highlight their respective strengths and trade-offs for production planning optimization.

2.1. Notation and Parameters

Model Parameters

b_t:	Inventory holding cost per unit in time period t.
c_t:	The cost of Backorders per unit within the time period t.
u_t:	The cost of a man's typical work hours during the period t.
v_t:	The cost of a man's extra work hours in time period t.
d_t:	Predicted demand for the period t.
h_t:	Hiring cost of single worker in time period t.
f_t:	Firing amount of a single employee in interval t.
μ:	Human hours required to produce a single unit of a product.
U_t:	Maximum number of regular work hours available for men during the interval
V_t:	Maximum overtime worker's hours for men during the period t.
I_0:	Initial Stock level.
ρ:	The regular workday into extraordinary hours is allowed.
T^-:	Number of planning periods in the horizon.

2.2. Mathematical Model: Variable Work Force Decision Variable:

P_t:	Production quantity in period t.
Q+:	Inventory carried at the end of period t.
B−:	Backordered amount in interval t.
S_t:	Net stock within time interval t
RE_t:	Regular hours of employment are utilized throughout the interval t.
OEt:	Overtime hours of labor used throughout the period t.
X_t:	Total employees who were hired during time period t.
Y_t:	Total employees who were fired during time period t.

Objective Function

$$\min \sum_{t=1}^{T} (a_t P_t + b_t Q_t^{+} + c_t B_t^{-} + u_t RE_t + v_t OE_t + h_t X_t + f_t Y_t) \quad (1)$$

The main objective is to maximize the profit and minimize the overall cost, which includes stock holding amount, aggregate production values and employment price (regular and over- time).

Model Constraints

Ensures that production, inventory, and backorders together satisfy the demand in each period.

$$P_{i,t} + S_{i,t-1}^{+} - S_{i,t}^{+} + S_{i,t}^{-} - S_{i,t-1}^{-} = d_{i,t} \quad \forall i, t \quad (2)$$

Links production quantity to the total regular and overtime man-hours required.

$$\mu P_t - (RE_t + OE_t) = 0 \qquad \forall t \quad (3)$$

Tracks the change in regular labour man-hours based on hiring and firing decisions.

$$RE_t - RE_{t-1} - X_t + Y_t = 0 \qquad \forall t \quad (4)$$

Restricts overtime man-hours to a fraction p of regular labour man-hours.

$$OE_t - p\,RE_t \leq 0 \qquad \forall t \qquad (5)$$

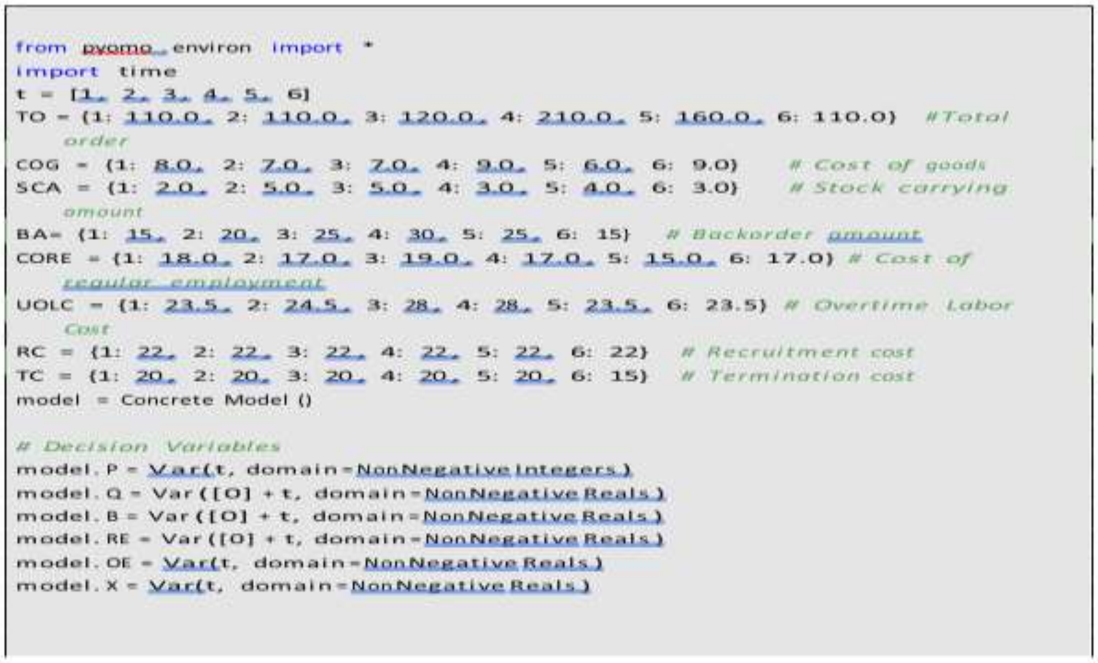
3. Algorithm Approach for solving the problem

Python solvers SciPy and Pyomo are used to perform the computational experiments on the numerical dataset shown in Table 1 [15]. Each iteration is run on a Windows 10 Pro Hp PROBOOK with 8 GB of RAM using the Python Community Edition 3.12 zone. One of the technical features is an Intel(R) Core (TM) i5-7200U processor with a base frequency of 2.50 GHz and a maximum turbo frequency of 2.71 GHz. This arrangement has been found to be enough for solving the global production planning model with changeable main-of-action and effectively comparing the solver's performances.

Table 1. Input Data for Variable Workforce Aggregate Production Planning Model

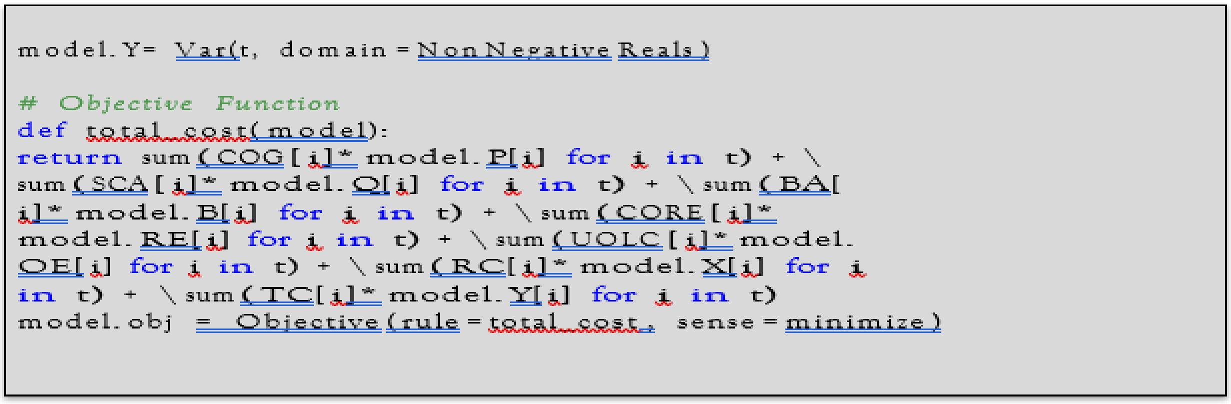
Month	January	February	March	April	May	June
Total Order	110.0	110.0	120.0	210.0	160.0	90.0
Cost of Goods	8.0	7.0	7.0	9.0	6.0	9.0
Stock Carrying Amount	2.0	5.0	5.0	3.0	4.0	3.0
Cost of Regular Employment	18.0	17.0	19.0	17.0	15.0	17.0
Overtime Labor Cost	23.5	24.5	28	28	23.5	23.5
Available Man-hours (Regular)	130	140	150	160	110	110
Available Man-hours (Overtime)	40	50	40	40	40	40
Backorder Amount	15	20	25	30	25	15
Cost of Hiring Workers	22	22	22	22	22	22
Cost of Firing Workers	20	20	20	20	20	15

1.1. Python Codes Pyomo Code



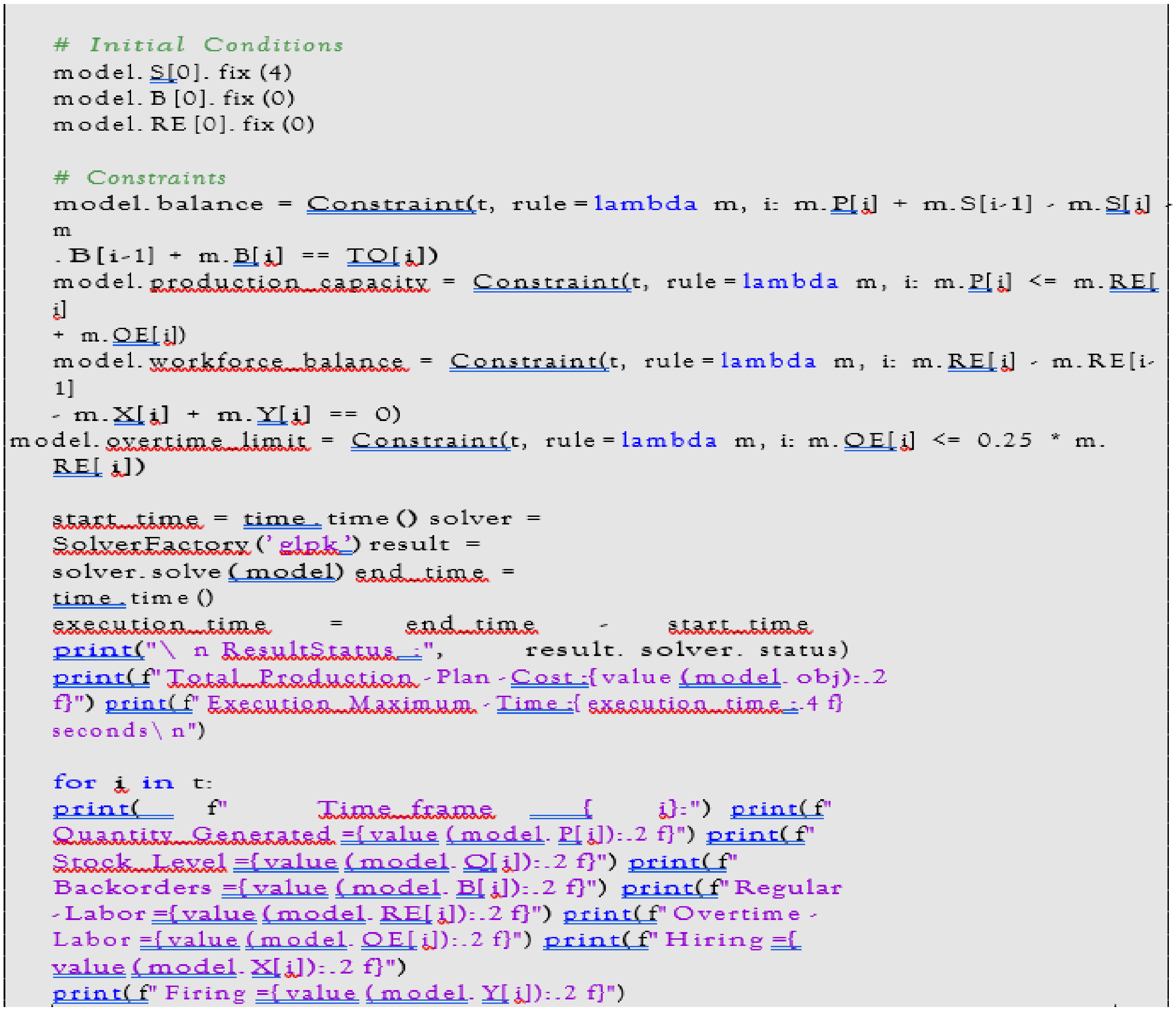
```
from pyomo. environ import *
import time
t = [1, 2, 3, 4, 5, 6]
TO = {1: 110.0, 2: 110.0, 3: 120.0, 4: 210.0, 5: 160.0, 6: 110.0}  #Total
    order
COG = {1: 8.0, 2: 7.0, 3: 7.0, 4: 9.0, 5: 6.0, 6: 9.0}    # Cost of goods
SCA = {1: 2.0, 2: 5.0, 3: 5.0, 4: 3.0, 5: 4.0, 6: 3.0}    # Stock carrying
    amount
BA= {1: 15, 2: 20, 3: 25, 4: 30, 5: 25, 6: 15}   # Backorder amount
CORE = {1: 18.0, 2: 17.0, 3: 19.0, 4: 17.0, 5: 15.0, 6: 17.0} # Cost of
    regular employment
UOLC = {1: 23.5, 2: 24.5, 3: 28, 4: 28, 5: 23.5, 6: 23.5} # Overtime Labor
    Cost
RC = {1: 22, 2: 22, 3: 22, 4: 22, 5: 22, 6: 22}   # Recruitment cost
TC = {1: 20, 2: 20, 3: 20, 4: 20, 5: 20, 6: 15}   # Termination cost
model = Concrete Model ()

# Decision Variables
model.P = Var(t, domain=NonNegative Integers)
model.Q = Var ([0] + t, domain=NonNegative Reals)
model.B = Var ([0] + t, domain=NonNegative Reals)
model.RE = Var ([0] + t, domain=NonNegative Reals)
model.OE = Var(t, domain=NonNegative Reals)
model.X = Var(t, domain=NonNegative Reals)
```



```
model.Y= Var(t, domain = NonNegativeReals)

# Objective Function
def total_cost(model):
return sum(COG[i]* model.P[i] for i in t) + \
sum(SCA[i]* model.Q[i] for i in t) + \sum(BA[
i]* model.B[i] for i in t) + \sum(CORE[i]*
model.RE[i] for i in t) + \sum(UOLC[i]* model.
OE[i] for i in t) + \sum(RC[i]* model.X[i] for i
in t) + \sum(TC[i]* model.Y[i] for i in t)
model.obj = Objective(rule = total_cost, sense = minimize)
```



```
    # Initial Conditions
    model.S[0].fix (4)
    model.B [0].fix (0)
    model.RE [0].fix (0)

    # Constraints
    model.balance = Constraint(t, rule = lambda m, i: m.P[i] + m.S[i-1] - m.S[i] -
    m
    .B[i-1] + m.B[i] == TO[i])
    model.production_capacity = Constraint(t, rule = lambda m, i: m.P[i] <= m.RE[
    i]
    + m.OE[i])
    model.workforce_balance = Constraint(t, rule = lambda m, i: m.RE[i] - m.RE[i-
    1]
    - m.X[i] + m.Y[i] == 0)
model.overtime_limit = Constraint(t, rule = lambda m, i: m.OE[i] <= 0.25 * m.
    RE[ i])

    start_time = time.time() solver =
    SolverFactory('glpk') result =
    solver.solve(model) end_time =
    time.time()
    execution_time     =     end_time     -     start_time
    print("\ n ResultStatus :",        result. solver. status)
    print(f"Total Production - Plan - Cost :{value (model. obj):.2
    f}") print(f" Execution Maximum - Time :{ execution_time :.4 f}
    seconds\ n")

    for i in t:
    print(     f"        Time frame      {       i}:") print(f"
    Quantity Generated ={value (model. P[ i]):.2 f}") print(f"
    Stock Level ={value (model. Q[ i]):.2 f}") print(f"
    Backorders ={value (model. B[ i]):.2 f}") print(f"Regular
    -Labor ={value (model. RE[ i]):.2 f}") print(f"Overtime -
    Labor ={value (model. OE[ i]):.2 f}") print(f"Hiring ={
    value (model. X[ i]):.2 f}")
    print(f" Firing ={value (model. Y[ i]):.2 f}")
```

SciPy code

Same data given in table 1 is used to perform APP problem but here SciPy is the solver.

```
import numpy as np
from scipy.optimize import linprog
import time# <-- For tracking execution time

# Periods
t = [1, 2, 3, 4, 5, 6]
n = len(t)

# Parameters
TO = [110.0, 110.0, 120.0, 210.0, 160.0, 110.0]
COG = [8.0, 7.0, 7.0, 9.0, 6.0, 9.0]
SCA= [2.0, 5.0, 5.0, 3.0, 4.0, 3.0]
  BA    [15,    25, 30,     15]
   =     20,    25,
  CO       =    17.0,       17.0, 15.0,
  RE  [18.0,    19.0,       17.0]
   U       =    24.5, 28    , 23.5,
  OL  [23.5,    , 28        23.5]
   C
  RC    [22,    22, 22,     22]
   =     22,    22,
  TC    [20,    20, 20,     15]
   =     20,    20,

block = n
Nvars = 7 * block

# Objective function
c = np.array(COG + SCA + BA + CORE + UOLC + RC + TC)

bounds = [(0, None)] * Nvars

# Equality constraints
A_eq = []
b_eq = []

for i in range(n):
row = np.zeros(Nvars) row[i]
= 1
if i == 0:
Itm1 = 4
Bt0 = 0
else:
row[block + (i - 1)] = 1 row[
block * 2 + (i - 1)] = -1
```

```
  A_eq . append ( row )
  b_eq . append ( TO[ i] - Itm1 + Bt0 if i == 0 else TO[ i])

  # Regular labor change constraints
  for i in range ( n):
  row = np. zeros( Nvars) row [
  block * 3 + i] = 1 if i > 0:
      row [ block * 3 + i - 1] = -1 row [
  block * 5 + i] = -1 row [ block
  * 6 + i] = 1 A_eq . append ( row )
  b_eq . append (0)

  # Inequality constraints
  A_ub = []
  b_ub = []

  for i in range ( n):
  row = np. zeros( Nvars) row [ i]
  = 1
  row [ block * 3 + i] = -1 row [
  block * 4 + i] = -1 A_ub .
  append ( row ) b_ub . append
  (0)

  for i in range ( n):
  row = np. zeros( Nvars) row [
  block * 4 + i] = 1 row [ block *
  3 + i] = -0.25 A_ub . append (
  row ) b_ub . append (0)

  # === Solve and Time ===
  start_time = time . time ()

result = linprog ( c=c,
  A_ub =A_ub , b_ub
  =b_ub , A_eq =A_eq ,
  b_eq =b_eq , bounds=
  bounds , method =' highs
  '
  )
```

```
end_time = time.time()

execution_time = end_time - start_time



# === Output ===

print (" Solution    Status :",    result. message ) print(f"
Execution Time: {execution_time :.6f} seconds")



if result. success: x =
    result. x

    print(f"Total Production Plan Cost = {result. fun :.2f}") Pt =
    x[0: block]

    Qt = x[block: block * 2]

    Bt = x[block * 2: block * 3] REt =
    x[block * 3: block * 4] OEt = x[
    block * 4: block * 5] Xt = x[
    block * 5: block * 6] Yt = x[
    block * 6: block * 7]



    for i in range(n):

        if Pt[i] > 0: print(f"Quantity_Produced_{i + 1} = {Pt[i]:.2f}") if
        Qt[i] > 0: print(f"Inventory_{i + 1} = {Qt[i]:.2f}")
```

2. Results and Discussion

A comparison between a relaxation in linear programming (LP) and a mixed-integer (MIP) approach to programming has been made in order to assess the performance of various solvers for the APP model. While the relaxation in LP was resolved with SciPy to obtain a faster and almost optimal solution, the MIP model was implemented in Pyomo to obtain an exact result. The results are summarized in the following table 2.

TABLE 2. MIP and LP relaxation comparision for the variable workforce APP model

Solver	Method	Optimal Cost	Execution Time (s)
Pyomo	MIP	24382	0.1183
SciPy	LP Relaxation	24380.47	0.00709

Figure 1 presents a bar graph comparing the performance of the Pyomo (MIP) and SciPy (LP relaxation) solvers for the variable workforce aggregate production planning model. The left axis shows the optimal cost obtained by each solver, while the right axis represents execution time in seconds. The bar graph allows for a clear side-by-side comparison, highlighting that both solvers achieve nearly identical costs, but SciPy requires significantly less computation time.

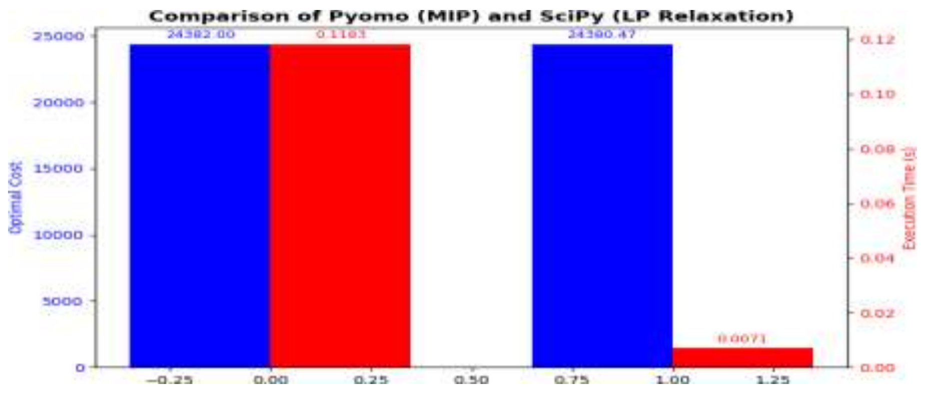


FIGURE 1. Comparison of Pyomo (MIP) and SciPy (LP relaxation)

Figure 2 shows a line graph illustrating the same comparison with two separate y-axes: one for optimal cost and another for execution time. The plotted points are connected to emphasize the difference in trends for cost and time. This visual representation makes it easier to observe that while the cost values are almost equal, the execution time difference between the two solvers is substantial.

3. Conclusion

The comparison reveals a balance between calculation speed and accuracy. Although it requires more significant computation time, mixed-number programming ensures precise answers that strictly meet integrity constraints. On the other hand, by allowing fractional decision vari- ables, linear programming relaxation yields nearly optimal results much more quickly. This compromise implies that LP relaxation is well suited to quick approximations or large-scale scenario testing, while mixed-number programming is still the preferred option when solution accuracy and mathematical viability are crucial to a practical implementation.

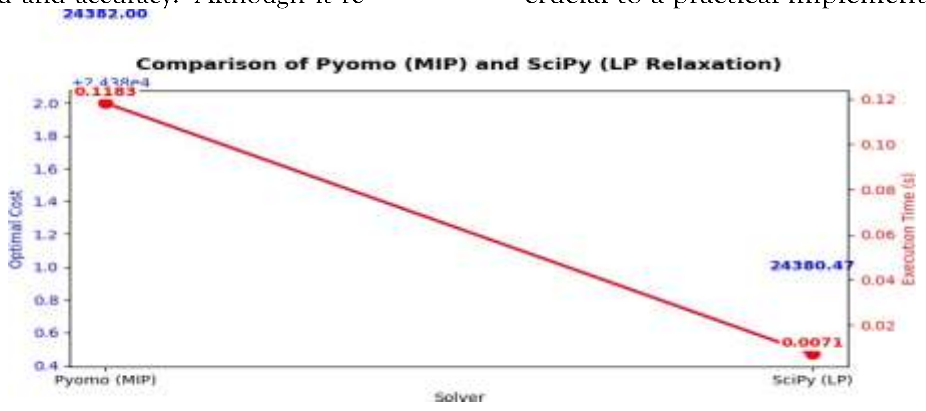


FIGURE 2. Comparison of Pyomo (MIP) and SciPy (LP relaxation)

References

Diwekar UM. Introduction to applied optimization. Springer Nature; 2020 Oct 29.

Kou G, Liu X, Peng Y, Shi Y, Wise M, Xu W. Multiple criteria linear programming approach to data mining: Models, algorithm designs and software development. Optimization Methods and Software. 2003 Aug 1;18(4):453-73.

Vo S, Woodruff DL. Introduction to computational optimization models for production planning in a supply chain. Springer Science & Business Media; 2006 Mar 20.

Silva Filho OS, Cezarino W, Ratto J. Aggregate production planning: Modeling and solution via Excel spreadsheet and solver. IFAC Proceedings Volumes. 2010 Jan 1;43(17):89-94.

Souza GC. Supply chain analytics. Business Horizons. 2014 Sep 1;57(5):595-605.

Gholamian N, Mahdavi I, Tavakkoli-Moghaddam R. Multi-objective multi-product multi-site aggregate production planning in a supply chain under uncertainty: fuzzy multi-objective optimisation. International Journal of Computer Integrated Manufacturing. 2016 Feb 1;29(2):149-65.

Anand Jayakumar A, Krishnaraj C, Nachimuthu A. Aggregate production planning: Mixed strategy. Pak.
J. Biotechnol. 2017;14(3):487-90.

Biazzi JL. Aggregate planning for probabilistic demand with internal and external storage. Journal of Operations & Supply Chain Management (JOSCM). 2018 Jun 15;11(1):37-52.

Demirel E, zelkan EC, Lim C. Aggregate planning with flexibility requirements profile. International Journal of Production Economics. 2018 Aug 1;202:45-58.

Cheraghalikhani A, Khoshalhan F, Mokhtari H. Aggregate production planning: A literature review and future research directions. International Journal of Industrial Engineering Computations. 2019 Apr 1;10(2):309-30.

Jamalnia A, Soukhakian MA. A hybrid fuzzy goal programming approach with different goal priorities to aggregate production planning. Computers & Industrial Engineering. 2009 May 1;56(4):1474-86.

Parganiha K. Linear programming with python and pulp. International Journal of Industrial Engineering Research and Development. 2018 Sep;9(3):01-8.

Sodero A, Jin YH, Barratt M. The social process of Big Data and predictive analytics use for logistic & supply chain management. International Journal of Physical Distribution & Logistics Management. 2019 Aug 30;49(7):706-26.

Shete K, Shinde P, Tanpure P, Gunjal A. Analytics for Manufacturing Decisions in Supply Chain Manage-ment. International Journal for Research in Applied Science & Engineering Technology. 2019;7(V):861-7.

Schmid J, Teichert K, Chioua M, Schindler T, Bortz M. IndustryDriven Rapid Prototyping for Comparing Nonlinear Control Schemes in Pyomo. Chemie Ingenieur Technik. 2020 Dec;92(12):2016-27.

Rehman HU, Ahmad A, Ali Z, Baig SA, Manzoor U. Optimization of Aggregate Production Planning Problems with and without Productivity Loss using Python Pulp Package. Management and Production Engineering Review. 2021;12(4).

ALRIDHA AH, Salman AM, Al-Jilawi AS. Numerical Optimization Approach for Solving Production Plan- ning Problem Using Python language. Central Asian Journal of Mathematical Theory and Computer Sci- ences. 2022;3(6):6-15.

Patrick WM, Anselemo PI, Ronoh R, Mbugua S. Impact of predictive analytics of big data in supply chain management on decision-making. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. 2022 Jul;3307:225-38.

Esteso A, Boza A, Alemany MM, Gomez-Gasquet P. Conceptual Framework for Optimization Models in Industry 4.0 Context: Application to Production Planning. InIndustry 4.0: The Power of Data: Selected Papers from the 15th International Conference on Industrial Engineering and Industrial Management 2023 Jul 8 (pp. 119-127). Cham: Springer International Publishing.

Al Bashar M, Taher A, Johura FT. Utilizing Predictive Analytics for Enhanced Production Planning and Inventory Control in the US Manufacturing Sector. International Research Journal of Modernization in Engineering Technology and Science. 2024 Jun;6:8332-9.