Improving Chicken Breast Production Scheduling through Pareto Optimization with VB.NET

Addison Standridge (Team Leader), Carter Baldwin, Kevin Connelly, Joshua Smith
Industry Partners: Turner Vance, Director Continuous Improvement, Ross Vandever, VP Continuous Improvement, Kerry Bartholomew, Senior Continuous Improvement Manager
May 1, 2024

Simmons Prepared Foods

Headquartered in Siloam Springs, Arkansas, Simmons Foods, Inc. and its associated enterprises are primary providers of poultry, pet, and animal nutrition items. Simmons Prepared Foods ranks among the top 15 poultry producers in the United States and operates as a division of Simmons Foods within the foodservice industry. Our system of interest is the chicken breast scheduling process at their Van Buren Plant.

Current Scheduling Process

Simmons Prepared Foods currently schedules customer orders based on employee knowledge. The master schedule creates a schedule of all the orders for each plant, and then the plant scheduler assigns orders to shifts while trying to minimize tardiness. After this, a plant manager instructs machines and people to decide what order to run the products in on a single shift. Constant changes in customer specifications, chicken supply, and throughputs lead to updates to the schedule to attempt to satisfy all orders in a timely fashion. Without long term employee knowledge, there would be no process or person in place that can schedule Simmons' orders effectively.

		Monday				
		2/20/23		Plant Manager		303
		WIP CODE	WHERE USED	QTY	OLDEST KILL	DATE USED
DB20	1ST SHIFT	1T/K4095110H	VCP	32,456	15-Feb	20-Feb
		1T/V4115140H	VBW	14,573	15-Feb	20-Feb
		1T/V/K4085095H	VCP	1,396	20-Feb	25-Feb
		1T/V4100123H	VBS	8,294	16-Feb	21-Feb
	2ND SHIFT	1T/K4105125H	VBS	10,648	16-Feb	21-Feb
		1T/V4145160H	VBW	19,567	16-Feb	21-Feb
		1V/K4135215H	VCP	28,345	16-Feb	21-Feb

Multiple Schedules Through Pareto Optimization

We created an optimization model that assigns customer orders to shifts through pareto optimization with an objective that minimizes trim and an epsilon constraint that controls the allowable tardiness limit for orders after the first week.

Parameters

d_j : due date of customer order *j*
m_j : demand by order *j*
n_i : inventory at time of shift *i*
r_j : median size of filet of order *j*
s_i : median size of breasts of shift *i*
p_j : throughput of order *j*
c_i : capacity to perform work of shift *i*
f_i : day of shift *i*
n_j : nugget/filet distinction for order *j*
l : tardiness limit

Indexes

I: shift
J: job

Decision Variables

X_{ij} : 1 if order *j* is worked during shift *i*, 0 otherwise
T_j : tardiness of job *j* in days

Model

$$\sum_{j=1}^{J}\sum_{i=1}^{I} X_{ij}(s_i - r_j) * n_j$$

Objective function that minimizes the difference between the median sizes of breasts on shifts i..I and on orders j..J

$$\sum_{j=1}^{J} X_{ij} \geq 1$$

Each order assigned to a shift

$$\sum_{j=1}^{J} X_{ij} * p_j \leq c_i$$

Capacity for each shift

$$\sum_{j=1}^{J} X_{ij} * m_j \leq n_i$$

Inventory for each shift

$$T_j \geq 0$$

For every job j, the tardiness limit does not benefit from being negative

$$s_j \geq r_j * X_{ij}$$

For every shift i and every job j, a filet cannot be cut out of a breast that is smaller than itself

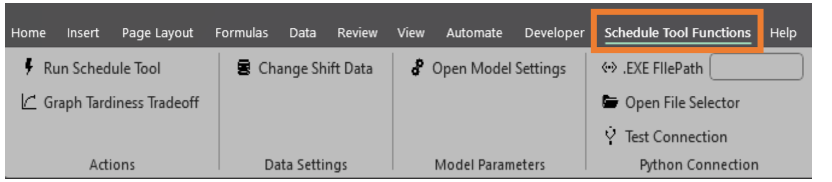
$$\sum_{i=1}^{I} X_{ij} * (f_i - d_j) \leq T_j$$

For every shift i and every job j, assign tardiness parameter

$$\sum_{j=1}^{J} \{d_j \leq 5\} T_j * 1000 + \sum_{j=1}^{J} \{d_j > 5\} T_j \leq l$$

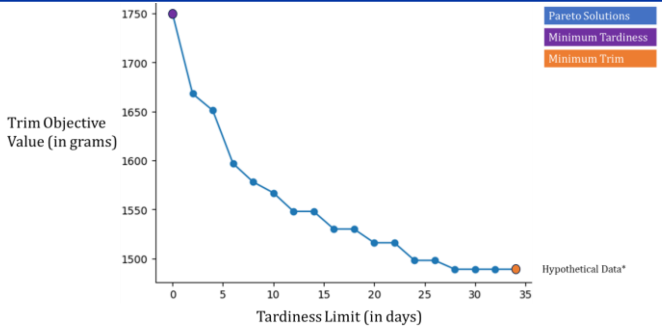
Epsilon constraint that allows us to control total allowable days tardy and ensure all orders worked within a week are on time

Integrated Excel Add-In



Automating Simmons current scheduling sheet by leveraging Microsoft's VB.NET along with python libraries such as pandas and OR-Tools allows for the plant scheduler to download our optimization model and input real-time data to produce multiple schedules. Additionally, our tool has user-friendly forms to input data not included in the scheduling sheet, as well as options for how many schedules to produce.

Pareto Optimization Results



A graphical output made from python script is used to show the lowest trim value for every tardiness limit availability. Simmons expects for this tool to significantly decrease the time spent scheduling, while also decreasing the amount of trim they produce.