Contents

List of Tables

List of Figures

1 Introduction

The problem of *effectively* allocating finite resources has been tackled for centuries. Certainly, as long as there have been functional economies, even systems as simple as barter networks, producers have been pressured to utilize labour, capital, and raw materials in the most efficient manner possible. More generally, there exist a class of problems known as Assignment Problems.

In the most general case, the Assignment Problem boils down to "essentially [trying] to address the issue of getting the best from every [resource] allocation" [Pai, 2012, p. 8]. This could mean assigning workers of various skill levels to machines in a factory, scheduling specific types of products to be produced on assembly lines of varying capacity, reducing change-over and re-tooling costs during production, or many other problems. In this paper, we examine the problem of assigning a varied workload of tasks to a group of employees of varying skill levels in such a way that we minimize the elapsed time, commonly referred to as the MAKESPAN.

Our problem is important because it directly affects the throughput and, therefore, the profitability of the office we study. By finding an optimal assignment strategy that minimizes the amount of time it takes to complete a task, we can be sure that the office operates in an efficient, cost-effective, and sustainable manner over the long term.

1.1 Historical Context

While people have encountered Assignment Problems for at least several centuries, they lacked the tools to effectively tackle them, especially in the general sense, until fairly recently. Indeed, neither the Renaissance nor the Industrial Revolution produced the necessary confluence of applied and theoretical mathematics and advanced computational technology required to enable business owners to solve these types of problems unassisted.

By the mid-20th century, however, a number of critical breakthroughs would begin to bring powerful mathematical techniques to the masses. Alan Turing had worked to design high-performance code-breaking analog computers and had invented what we now call software while "attacking problems of calculability and Hilbert's Entsheidungsproblem"[Petzold, 2008, p. 65]. John Von Neumann envisioned the architecture of the modern digital computer. Wassily Leontif had "opened the door to a new era in mathematical modelling [sic]"[Lay, 2006, p. 1], work for which he would be "awarded the 1973 Nobel Prize in Economic Science" [Lay, 2006, p. 1]. And a secret group colloquially referred to as "Blackett's Circus" had created a new industry while using science and mathematics to defeat the Nazis. They called it OPERATIONS RESEARCH.

Within the British Admiralty, and reporting directly to Winston Churchill, a scientist and Navy officer named Patrick Blackett led the "operations research group, ...a subdepartment of the navy's Scientific Research and Experiment Department." [Budiansky, 2013, p. 462] While developing technologies like RADAR and pioneering the application of science to military strategy, Blackett's team laid the foundation for the post-war surge in operations research work. Such work would eventually lead to minor business revolutions like lean manufacturing and its offshoot, agile software development.

It was during these decades of rapid technological advancement that the mathematical modelling techniques used in this paper were refined. Fortunately for the authors, we now live in a time where complex mathematical models can be effortlessly solved with even a mediocre personal computer.

2 The Employee Scheduling Problem

In this paper, we examine an insurance office that handles two types of work: creating new policies, and preparing claims. The office has three employees of varying skill levels, who each take different lengths of time (on average) to complete a given task. Here we see the average time (in minutes) required for each employee to complete each task type.

	New Policy	Claim
Employee 1	10	28
Employee 2	15	22
Employee 3	13	18

Tab. 1: Average task processing time in minutes.

The insurance company has compiled the timings above by empirical measurement. We assume that the company has a healthy business and, therefore, that there is a steady stream of New Policy and Claim tasks that require processing by the office employees. Under these circumstances, the insurance company is interested in assigning some fraction of the incoming tasks to each employee so as to minimize the makespan, or the elapsed time required to finish all of the tasks, while maintain the ratio of New Policy and Claim constant.

2.1 Modelling Approach

To answer the question "how should the insurance company assign tasks to its employees to minimize the makespan?", we rely on a linear programming technique known as Simplex Minimization. We consider p_i and q_i, the fraction of New Policy tasks and the fraction of Claim tasks, respectively, to be assigned to the i^{th} employee.

Our objective function is simply t, the elapsed time (in minutes) required to complete all of the tasks which needs to be minimized.

2.1.1 Parameters

We consider all of the average processing times listed in Table 1 to be model parameters.

2.1.2 Variables

We formally define the model variables as follows:

p_1, p_2, p_3	The fraction of New Policy work assigned to each employee
q_1, q_2, q_3	The fraction of Claim work assigned to each employee

Tab. 2: Basic Model Variables

2.1.3 Constraints

Since we are dealing with human employees and real-world tasks, we assume all trivial and non-negative constraints,q_i,t,$p_i \geq 0$. For example, it would be rather foolish to assign a negative number of tasks to an employee for completion. Here we also define the non trivial constraints.

$p_1 + p_2 + p_3 = 1$	We must assign exactly all New Policy tasks
$q_1 + q_2 + q_3 = 1$	We must assign exactly all Claim tasks
$10p_1 + 28q_1 \leq t$	Ensure minimal makespan
$15p_2 + 22q_2 \leq t$	Ensure minimal makespan
$13p_3 + 18q_3 \leq t$	Ensure minimal makespan

Tab. 3: Basic Model Constraints

The first two constraints ensure that all the tasks would be processed by am employee. These constrains ensure that we would not end up with the theoretical, trivial, unworthy solution $p_i = q_i = t = 0$. In addition, any solution that does not satisfy the first two constrains corresponds to the case where not all of the tasks are assigned to employees and thus an unassigned portion of tasks would not be completed. The remaining other three constraints ensure that no employee exceeds the time required to complete all the tasks.

2.1.4 The Objective Function

Stated more precisely in mathematical terms, our objective is to "choose values of p_i and q_i (listed in Table 2) that will minimize the objective function t, subject to all trivial constraints,q_i,t,$p_i \geq 0$, and all non-trivial constraints listed in Table 3." Since we are making no assumption on the number of policies and claims that needs to be processes, therefore, the objective function t is not the same as the time in minutes required to complete all tasks. In fact, since there is a steady, health, long (infinite) sequence of policies and claims that needs to be processed, therefore it would take infinite minutes to processes all the tasks. On the positive side, the objective t corresponds to the best task assignment strategy that would minimized this infinite time. For lack of a better word, we will refer to$t$ as "makespan," and "total time" through out this report.

3 Solving the Basic Employee Scheduling Model

The basic model described above is well-suited for the Simplex Minimization algorithm, and requires little modification. We set up and solved the model in a spreadsheet, seen below.

	Efficiency of Workers			
	Processing Time per Policy (min)	Processing Time per Claim (min)		
Employee 1	10	28		
Employee 2	15	22		
Employee 3	13	18		
	Decision Variables		Constraints	
	Percent of Policies Processed	Percent of Claims Processed	Employee's Processing *"Time"*	Total Processing *"Total Time"*
Employee 1	99.402985%	0.000000%	9.940298507	9.940298507
Employee 2	0.597015%	44.776119%	9.940298507	9.940298507
Employee 3	0.000000%	55.223881%	9.940298507	9.940298507
Total	100%	100%		
Objective function *t*			9.940298507	

Fig. 1: Basic Model Solution (Excel)

To download the spreadsheet for closer inspection, see Appendix A.
From the solution, we find that the optimal assignment strategy is

New Policy Assignments	Claim Assignments
$p_1 = 333/335 \approx 0.994$	$q_1 = 0$
$p_2 = 2/335 \approx 0.006$	$q_2 = 30/67 \approx 0.448$
$p_3 = 0$	$q_3 = 37/67 \approx 0.552$

Tab. 4: Optimal solution to the basic model

Put plainly, over the long term management should assign about 99.4% of New Policy tasks to Employee 1, about 0.6% of New Policy tasks and about 44.8% of Claim tasks to Employee 2, and about 55.2% of Claim tasks to Employee 3.

With this optimal solution, we find resulting optimal makespan is $t = \frac{666}{67} \approx$ 9.94 minutes or about 9:56.

3.1 Sensitivity Analysis

From a sensitivity analysis obtained using Excel's simplex Solver, we find, as follows, the shadow prices for our two most important constraints.

Constraint	Shadow Price
$p_1 + p_2 + p_3 = 1$	4.0299
$q_1 + q_2 + q_3 = 1$	5.9104

Tab. 5: Constraint shadow prices

We see that the basic model is slightly more sensitive to the constraint $q_1 + q_2 + q_3 = 1$, the fraction of Claim tasks assigned, since that constraint has the higher shadow price. We believe that this is caused because the average time it takes for any employee to handle a Claim task is higher than the average time to process a New Policy task. Thus, the insurance company may be able to streamline its operation by finding ways to reduce the time it takes to process a Claim task. Such strategies might include more employee training, process automation, or improvements in electronic record keeping systems

3.2 Observations of the Basic Solution

Our solution is only useful in the long-term where there is an infinite sequence of tasks to be processed. For example, it wouldn't make sense to assign 0.6% of a single task to one employee and most often, as . In addition to being impractical, the cost of context-switching as the employees handed off the partially completed task would outweigh any potential time savings. Rather, our scheduling solution would be better applied to a large list of tasks. For example, if the office had a long backlog of work (say, several hundred tasks) or if the insurance company is busy (and healthy), receiving a constant and fairly steady stream of work every day for the foreseeable future.

It is interesting to note that, while no *individual* employee is able to finish any single task in less than ten minutes, our optimal scheduling strategy has all tasks being completed in slightly less than 10 minutes per task, over the long term.

3.2.1 Criticism

This basic employee task appointment strategy has multiple flaws. Firstly, there is no information provided regarding the accuracy and methodology of the measured timings listed in Table 1. Additionally, a linear approximation of employee's average processing time (1) , cannot be an accurate representation of of real world worker productivity over a long period of time. For example, the authors believe that it is fairly unlikely that an employee's productivity would remain constraint throughout the day (let alone over weeks, months, or years). Our model does not take into account any aspect of productivity fluctuations caused by factors such as fatigue, boredom, enthusiasm, learning over course of time, etc.

Additionally, The model does not take into account the ratio of policies and claims that need to be processed. It is very likely that the average number of policies and claims that need to be processed annually are not the same. For example, it it likely that there would be more policies that need to be processed and fewer claims. To further dig into the probable complexity of this problem, we investigate some variation of this problem by introducing further or different constrains.

The authors also wonder whether such a scheduling strategy could foster animosity and politics between the employees. We leave this as an open question for the reader.

4 Enhanced Employee Scheduling Problems

The insurance company may be interested to learn how the Basic Solution reacts to other real-world conditions. Here, we present

4.1 Employee Sick Time

Suppose that one of the employees becomes ill and cannot work. How would that affect the Basic Model? Does it matter which employee becomes ill? The answers to these questions could help management determine which employees to target with extra preventative health and wellness benefits (perhaps free flu vaccinations, or similar).

We can examine this question by removing one employee at a time from the scheduling model, then running the Simplex Minimization algorithm again on the augmented model. This results in three new solutions (one for each ill employee) which can be compared to determine which ill worker has the greatest impact on overall office productivity. Here we see the solutions to each augmented model.

Sick Employee	p_i values	q_i values	t value
1	$p_2 = 1$, $p_3 = 0$	$q_2 = 3/40$, $q_3 = 37/40$	$333/20 \approx 16.65$
2	$p_1 = 1$, $p_3 = 0$	$q_1 = 4/23$, $q_3 = 19/23$	$342/23 \approx 14.870$
3	$p_1 = 1$, $p_2 = 0$	$q_1 = 6/25$, $q_2 = 19/25$	$418/25 \approx 16.72$

Tab. 6: Solutions to the Employee Sick Time Model

While sick time for any of the employees has a clear negative impact on the processing time (that is, the elapsed time, t, to process a task increases) we can spot some differences. Sick time by Employees 1 and 3 has roughly the same affect on the makespan, t, whereas when Employee 2 is ill we see that the overall impact is comparatively low. Thus, from the insurance company's point of view, if any one employee should get sick, it would be preferable that employee 2 becomes ill. Therefore, management could mitigate the risk, at the least cost, by offering only Employees 1 and 3 extra preventative health care coverage such as seasonal flu vaccinations.

These results also suggest that Employee 2 may be the least productive member of the team.

With these findings on hand, management would be able to confidently predict the adjusted company-wide productivity and adjust deadlines, if necessary, as soon as they get news that an employee has taken ill.

4.2 Employee Training

Oftentimes, companies train their employees to increase efficiency and productivity. Moreover, even though if unlimited resources were available it would have been better for the companies to train all their employees in all aspect of the corporation, but they often select a subset of the employees to train in a specialized aspect of the corporation. To model a similar problem for our fictional insurance company, we assume that the company could train only one of its employees and hope to reduce the total time needed to complete New Policy and Claim applications.

4.2.1 Employee Selection Criteria

Percent increased efficiency of a given employees depend on the learning curve of that individual and certainly it depend on the suitability of the training curriculum and the length of the training. The simplest scenario would be to assume that all the employees have the same linear learning curve and that the curriculum would be the same for all employees. As a result, it is reasonable to conclude that no matter which employee is chosen, the employee would become more efficient by the same percentage. Let's assume that an employee would become 10% more efficient upon completion of the training.

4.2.2 Methodology and Outcome

Mathematically, the solution to this problem is the same as solving the basic model after replacing for the enhanced task completion time of the trained employee. Hence, this methodology requires solving the basic model 3 times, once per employee, and selecting the employee who yields a better result for the training.

Employee Selected for Training	New Policy	Claims
1	$10/1.1$	$28/1.1$
2	$15/1.1$	$22/1.1$
3	$13/1.1$	$18/1.1$

Tab. 7: Revised task processing time (in minutes) for the selected employee

Upon creating a Maple routine that solved the basic model, we found that training employee 3 would yield the shortest time, t. In addition, upon the training completion, employee 1 would be assigned a 3.2% lesser New Policies while employee 2 would be assigned 5.3 times previously assigned New Policies.

Employee	p_i values	q_i values	t value
1	$p_1 = 1,\ p_2 = 0,\ p_3 = 0$	$q_1 = \frac{89}{3889},\ q_2 = \frac{1710}{3889},\ q_3 = \frac{2090}{3889}$	$\frac{37620}{3889} \approx 9.67$
2	$p_1 = \frac{333}{344},\ p_2 = \frac{11}{344},\ p_3 = 0$	$q_1 = 0,\ q_2 = \frac{159}{344},\ q_3 = \frac{185}{344}$	$\frac{1665}{172} \approx 9.68$
3	$p_1 = \frac{333}{346},\ p_2 = \frac{13}{346},\ p_3 = 0$	$q_1 = 0,\ q_2 = \frac{285}{692},\ q_3 = \frac{407}{692}$	$\frac{1665}{173} \approx 9.62$

Tab. 8: Task division and completion time upon an employee complete training, assuming the selected employee would be 10% more productive

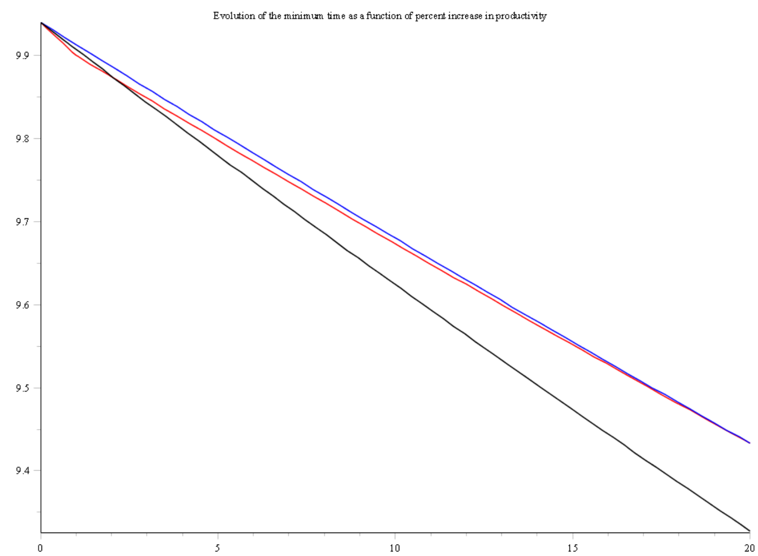

Fig. 2: Minimum time as a function of percent increased in productivity of one of the employees. Red, blue, and black represent workers 1, 2, and 3 respectively.

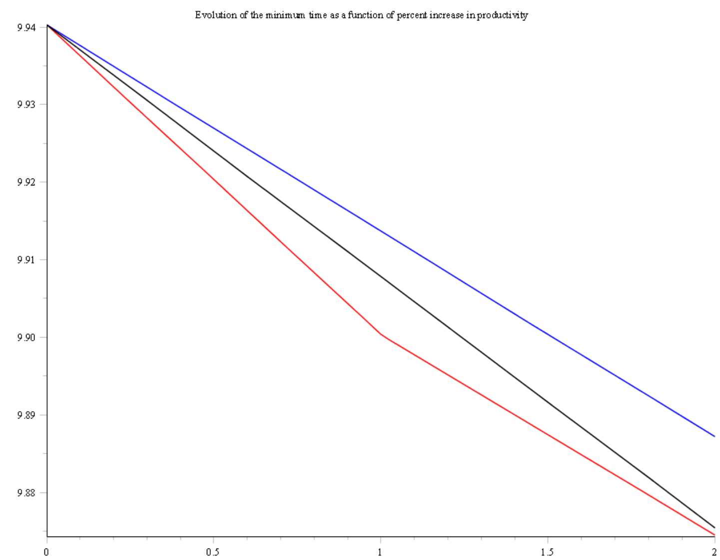

Fig. 3: Closer look at 0% to 2% increase in claimed productivity.

5 Conclusion

An enhanced model that would represent the real world should be a probabilistic model rather than deterministic that would take into account employee's sick and vacation time...

6 Appendix A

Rather than attach printed spreadsheets as an appendix, all of our project documents including spreadsheets, Maple source code, charts, and LaTeX files are available from our public repository, located at https://github.com/colefichter/math372_project_1.

References

[Budiansky, 2013] Budiansky, S. (2013). *Blackett's War: The Men Who Defeated the Nazi U-Boats and Brought Science to the Art of Warfare*. Knopf, Toronto.

[Lay, 2006] Lay, D. C. (2006). *Linear Algebra and Its Applications*. Pearson Education, Indianapolis, 3rd edition.

[Pai, 2012] Pai, P. P. (2012). *Operations Research: Principles and Practice*. Oxford University Press, New Delhi.

[Petzold, 2008] Petzold, C. (2008). *The Annotated Turing*. Wiley Publishing, Indianapolis.