Name: Folarin, Wasiu Junior

Matric Number: MTH/2014/014

Supervisor: Dr. B.S. Ogundare

September 2018

INTRODUCTION

In everyday life, we usually see the need to further classify a collection of items into non-overlapping, exhaustive sub-groups, with each group containing members satisfying a number of constraints. That is, membership in a group is based on a proof-positive check of each members against each of the constraining conditions.

Although, several methods in the field of Operations Research (OR), especially, have been developed to solve various assignment and optimization problems. Nonetheless, Linear Programming (a key tool of OR) is specifically defined thus:

> *"Linear Programming is a mathematical technique for determining the optimal allocation of resources and obtaining a particular objective(i.e., cost minimization or inversely profit maximization when there are alternative uses of the resources: Land, Labout, Capital, Materials, Machines, etc."*

The question therefore arises – what if our optimization doesn't involve minimizing cost of maximizing profit; what if we're just interested in

Allocation of objects to non-overlapping groups,

based on inate properties of the objects and constraining conditions

A REFERENCE USE CASE SCENARIO

Given that every intending NYSC corp member would be posted to a serving state based on the following criteria

- He/she should not be posted to his state of origin
- He/she should not be posted to his state of studying
- He/she can only be posted to a state with capacity to absorb him/her

Given these set of constraints how do we efficiently deploy each intending Corp member to a state?

THE ALGORITHM

$$Let\ U=\{u_{1,}u_{2,}u_3,\ldots,u_n\}$$

$$be\ a\ set\ of\ objects\ which\ are\ \text{to}\ be\ grouped\ into\ non-overlapping\ sets$$

$$Let\ P=\{p_1,p_2,p_3,\ldots,p_k\}$$

$$be\ k\ number\ of\ partitioning\ sets\ \text{to}\ which\ be\ the\ elements\ of\ the\ set\ U\ are\ \text{to}\ be\ uniquely\ assigned$$

$Also\ given\ are\ the\ following\ constraints$

1. $Each\ partitioning\ set, p_i\ has\ a\ maximum\ capacity\ of\ objects\ it\ can\ contain$

$$j(p_i)=V, where\ V\in\mathbb{N}$$

$Hence\ the\ total\ capacity\ of\ all\ partitions, p_i, i=1,2,3,\ldots,k$

$$X=\sum_1^k j(p_i)$$

And $so, we\ define\ the\ weight, w_i\ of\ each\ partition, p_i, thus$

$$w_i=\frac{j(p_i)}{\sum_i^k j(pi)}=\frac{j(pi)}{X}$$

$We\ now\ define\ the\ width\ of\ each\ partition\ p_i$ as

$$n(p_i)=|p_i|=w_i\cdot|U|=w_i\cdot n$$

2. $Inclusion/Exclusion\ Constraints, g_{i_k}, defined\ thus:$

$$g_{i_k}(u_j)=\begin{Bmatrix} c\in(0,1], & if\ u_j tests\ positive\ for\ partition\ k \\ c=0, & otherwise \end{Bmatrix}$$

$$where\ g_{i_k} is\ the\ g_i th\ constraint\ testing\ for\ compatibility\ of\ the\ u_j th\ object\ \text{in}\ the\ kth\ partition$$

$$where\ i\ is\ the\ counter\ for\ the\ total\ number\ of\ inclusion/exclusion\ constraints\ defined$$

3. *We now define the membership function for this algorithm thus*:

$$f_i(u_j)=\prod_{s=1}^{t} g_{i_s}(u_j)$$

$$where\ i=1,2,...,k;$$
$$j=1,2,...,n;$$
$$t=the\ total\ number\ of\ defined\ inclusion/exclusion\ constraints$$

4. *We now define the set* F_i

$$F_i=\{f_i(u_1),f_i(u_2),f_i(u_3),...,f_i(u_n) \qquad i=1,2,...,k\}$$

$$e.g.\ F_1=\{f_1(u_1),f_1(u_2),f_1(u_3),...,f_1(u_n)\}$$

5. *We now proceed* to *populate each partition*, p_i *with matching elements thus*:

$$Let\ F_{p_i}=F_i\setminus\bigcup_{j=1}^{i-1}\{f_j(u_x)\forall u_x\in p_j\}$$

$$Let\ Q_1=u_x \mid f_i(u_x)=max\{F_{p_i}\}$$

$$Q_2=u_x \mid f_i(u_x)=max\{F_{p_i}\setminus f_i(u_y) \mid u_y\in Q_1\}$$

$$\vdots$$

$$\Rightarrow Q_{|p_i|}=u_x \mid f_i(u_x)=max\{F_{p_i}\setminus\bigcup_{j=1}^{|p_i|-1}\{f_i(u_y) \mid u_y\in Q_j\}\}$$

$$\Rightarrow p_i=\bigcup_{j=1}^{|p_i|}\{Q_j\}$$

$$i=1,2,...,k$$

QED

NEXT STEPS

We shall proceed to verify this algorithm using a computer simulation of a concrete problem, and thereby investigate the proposed solution for correctness

THANK YOU!