We are planning to use warehouse location integer programming problem to optimize the resource allocation. This approach is generally used to represent the situation that an organization is considering a number of locations for building warehouses to supply its existing stores. Each warehouse has a n associated maintenance cost and its capacity in terms of maximum number of stores it can support. Each store can be supplied by exactly one warehouse and cost of supply to a store differs with the selected store.

This problem resembles with the task allocation and pool creation activity of ECRM. We are representing the user service requests as demands and the configuration of Pools as the location of the warehouse. In the first instance we took the constant values to represent the service request demand and the pool configuration. These values are ranging from 0 to 100. Both the service request demand and pool configuration are represented as (xi, yi) where x represents the CPU and y represents the memory.

A user service request R (xr, yr) can be allotted to a pool P(xp, yp) if:

xp-xr>=0 and yp-yr>=0 ----------- equation (1)

There may be an associated resource waste with the feasible allocation.

We define the resource waste as the difference of (xp-xr)+(yp-yr).

Following is the list of variables used in the problem formulation:

RPis a matrix of numRequests \* numpools dimensions, these are binary variables (0/1) where each row represents a service request and each column represents a pool. A value any element RPij with 1 value denotes that pool (Pj) can process this request (Ri) as it satisfies the condition represented in equation 1, otherwise the value will be 0.

A and P, are column vectors with numRequest rows, to represents the allocation and profit (respectively) of user service request processing. Values in column vector A are binary values (0/1) where value 1 denotes that the request is served by the system and the value 0 denotes that this request was rejected for the processing as there may be no space to occupy it. P values represents earnings from serving a job.

C is a row vector with numPools columns to represent the cost maintenance of each of the pools.

PC is another row vector with numPools columns to represent the capacity of each of the pools, in terms of number of requests.

Wrp is a matrix to represent the aggregated resource waste of the jobs allocation to pools. Considering user service request R (xr, yr) and a pool P (xp, yp), we define the resource waste Wrp of P serving request R as:

Wrp = (xp-xr) + (yp-yr).

We tested the formulation with two different objectives as:

1. The objective is to minimize the resource waste:
2. Another objective is to maximize the value, considering the profit of request allocation, cost of pool maintenance and resource waste. Consider the following variables:

A is a column vector (binary variable with 0/1 values to represent allocation, 1 means successful allocation and the value 0 denotes that this request was rejected for the processing as there may be no space to occupy it.).

P is another column vector representing the earnings from serving a job thus we call it as profit from each user service requests. Both A and P are having numRequests rows.

C is a row vector with numPools columns to represent the maintenance cost of each of the pools.

We define the value as the difference of the total profit and the total of cost and resource waste (P-C-W). The objective function to maximize the value is defined as:

Constraints: The above two different objectives are achieved under following to constraints.

1. Multiple pools may be eligible for a request allocation by satisfy the condition in equation 1 but there is a constraint that but only one of them should be allotted this request. This constraint is represented as:

=1

1. Another constraint is that the number of allotted requests to a pool should be less than or equal to the pool capacity. This is represented as: