Assignment 1

Resource Allocation in Edge Computing: A GAP based formulation

B. Resource allocation problem formulation

In this work, the resource allocation problem is a service placement problem that maximizes the satisfaction of users. The formulation is given as follows:

SE is the set of n ESPs, i.e., $SE = \{E_1, E_2, \ldots, E_n\}$, and SD is the set of m resource users or IoT devices, i.e., $SD = \{D_1, D_2, \ldots, D_m\}$. Each ESP $E_j \in SE$ has a maximum hardware capacity to offer. The requirement of Virtual Machine (VM) of each D_i from each ESP E_j may be different because of the location of ESPs, their network configuration, fixed types of VMs, etc., and so do the users' satisfaction. The objective of the allocation of resources is the maximization of satisfaction of resource users in such a way that the hardware capacity of each ESP, HC_j, is sufficient to satisfy the hardware resources occupied by the requested VMs of assigned users at the ESP.

Various quality parameters, such as computation power, response time, reliability, etc., play a role in measuring the satisfaction of a resource user. In utility theory, quality parameters are aggregated to define the degree of satisfaction, i.e., a user's utility. In edge computing, we can aggregate quality parameters to define the utility of the resource users. The importance of quality parameters may vary from one user to another user. Therefore, we use the weighted aggregated

utility function. Let SQ is the set of o quality parameters in edge computing, i.e., $SQ = \{q^1, q^2, \ldots, q^o\}$, q_{ij}^l is normalized value of l^{th} quality parameter that ESP E_j offers to the resource user D_i and w_i^l is preference of D_i for l^{th} quality parameter, modelling a dedicated service level agreement for each user. The utility of resource user D_i from the VM_{ij} running on ESP E_j satisfying the minimum requirement of D_i can be written as given in equation (1).

$$U_{ij} = \sum_{l=1}^{o} w_i^l \times q_{ij}^l, \qquad (1)$$

where, $\sum_{l=1}^{o} w_i^l = 1$.

Preferences assigned to quality parameters by the users reflect the significance of quality parameters for the users' requirements. For example, a user handling critical and sensitive applications gives a higher preference for trust, and a user running a computing-intensive application gives a higher preference for CPU. A user will decide the preference for quality parameters based on its application's requirements [7].

The objective of the service placement problem $\boldsymbol{SP}$ is the maximization of resource users' satisfaction. $\boldsymbol{SP}$ can be formalized as follows:

$$\boldsymbol{SP}: max \sum_{E_j \in SE} \sum_{D_i \in SD} x_{ij}\, U_{ij}, \tag{2}$$

such that,

$$\sum_{D_i \in SD} x_{ij} VM_{ij} \leq HC_j, \forall E_j \in SE, \tag{3}$$

$$\sum_{E_j \in SE} x_{ij} \leq 1, \forall D_i \in SD, \tag{4}$$

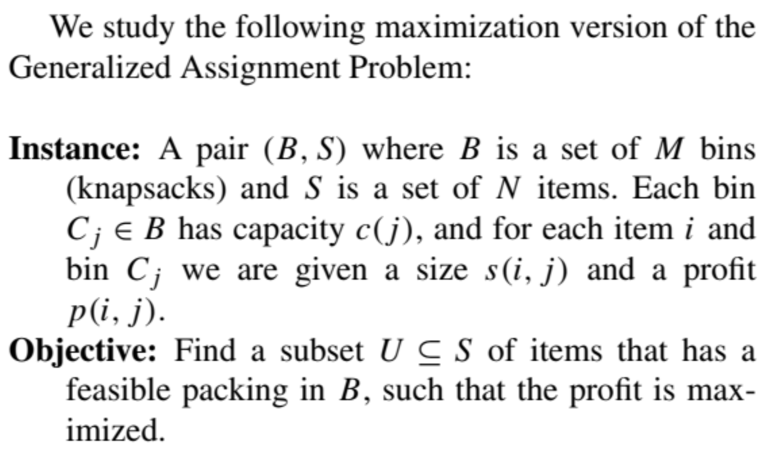
$$x_{ij} \in \{0,1\}, \forall D_i \in SD, \forall E_j \in SE. \tag{5}$$

Equation (2) is the objective function that ensures the maximization of satisfaction of resource users. Equation (3) is a capacity constraint, i.e., the hardware capacity of ESP must be greater than or equal to the resources occupied by VMs running on the server of ESP. Equation (4) guarantees that an ESP can serve more than one user, and one user gets the VM from only one ESP, subject to the availability of physical resources.

Theorem 1: The proposed service placement problem is NP-Hard.

Proof: To prove that the proposed service placement problem is NP-hard, we demonstrate that the proposed problem is similar to the Generalized Assignment Problem (GAP) [31]. In GAP, there are multiple knapsacks and multiple items. The objective is to assign each item to at most one knapsack in such a way that it maximizes the total profit and satisfies the capacity constraint, as each knapsack has a fixed capacity. Here, the knapsack in which the item is placed decides the profit of placing the item. In the proposed problem, multiple ESPs with fixed capacity and multiple users exist. VM's capacity and utility may vary for the same user from one ESP to another. The problem's objective is to maximize user satisfaction and satisfy the capacity constraint as in equation (3). The proposed problem can be mapped into a GAP problem (an NP-hard problem), making the proposed service placement problem an NP-hard problem. ■

GAP Problem definition

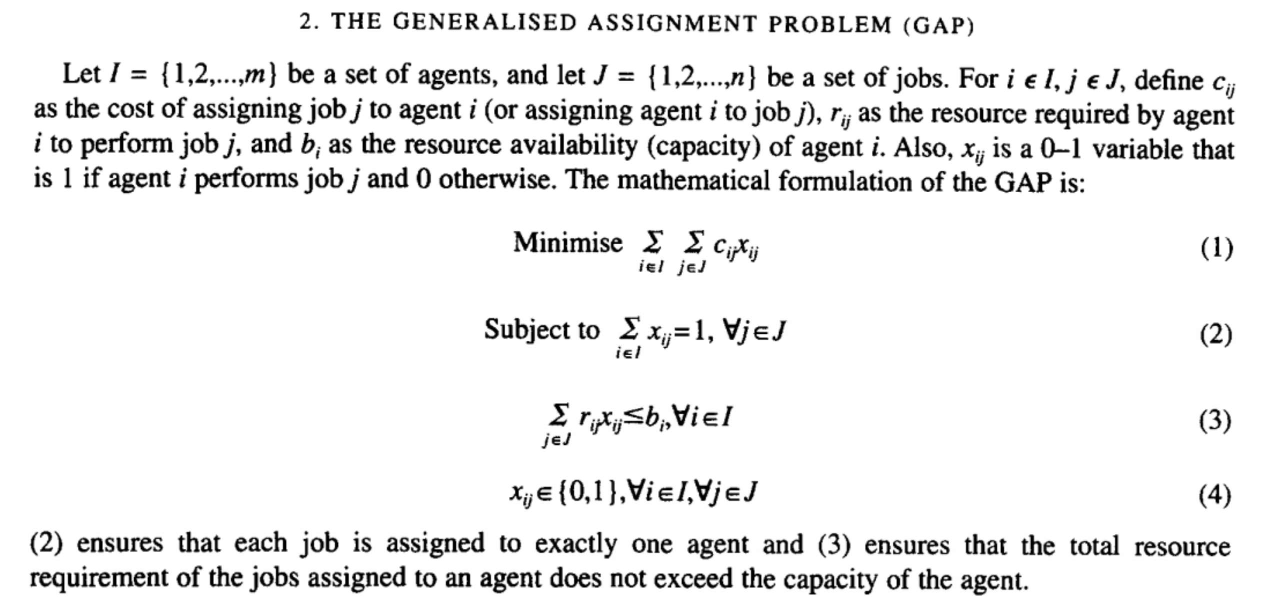


We study the following maximization version of the Generalized Assignment Problem:

Instance: A pair (B, S) where B is a set of M bins (knapsacks) and S is a set of N items. Each bin $C_j \in B$ has capacity $c(j)$, and for each item i and bin C_j we are given a size $s(i, j)$ and a profit $p(i, j)$.

Objective: Find a subset $U \subseteq S$ of items that has a feasible packing in B, such that the profit is maximized.

Another formulation for GAP Problem



2. THE GENERALISED ASSIGNMENT PROBLEM (GAP)

Let $I = \{1,2,...,m\}$ be a set of agents, and let $J = \{1,2,...,n\}$ be a set of jobs. For $i \in I, j \in J$, define c_{ij} as the cost of assigning job j to agent i (or assigning agent i to job j), r_{ij} as the resource required by agent i to perform job j, and b_i as the resource availability (capacity) of agent i. Also, x_{ij} is a 0–1 variable that is 1 if agent i performs job j and 0 otherwise. The mathematical formulation of the GAP is:

$$\text{Minimise} \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \qquad (1)$$

$$\text{Subject to} \sum_{i\in I} x_{ij}=1, \forall j\in J \qquad (2)$$

$$\sum_{j\in J} r_{ij}x_{ij}\le b_i, \forall i\in I \qquad (3)$$

$$x_{ij}\in\{0,1\}, \forall i\in I, \forall j\in J \qquad (4)$$

(2) ensures that each job is assigned to exactly one agent and (3) ensures that the total resource requirement of the jobs assigned to an agent does not exceed the capacity of the agent.

1. Find the optimal value for the GAP-based Resource Allocation problem mentioned above and verify the optimal fitness value for each instance in the GAP dataset (GAP1–GAP12). Additionally, plot a graph representing the optimal fitness value for each instance. Description and link to the dataset is given below.

   - The dataset files are available at https://people.brunel.ac.uk/~mastjjb/jeb/orlib/gapinfo.html. Description of these datasets is given in appendix 1.
   - Sample Data set is given below:
     1 – Number of problem set

3 4 – number of servers, number of users
Utility of allocating server j to user i (j=1,...,n) (between 0 and 1)
3 1 5 25
1 1 15 15
5 1 25 5
Number of VM allocated to different users from each server
1 2 2 1
1 3 3 2
1 3 4 3
2 3 4 – resource capacity of each server