Two enhancements to a flexible pricing control system for allocating computer resources

R. W. Zmud

School of Management, Clarkson College of Technology, Potsdam, New York 13676, USA

Two enhancements to an earlier presented flexible pricing for allocating computer resources are presented. The first incorporates a service quality dimension within the control system and results in a pricing system that directly influences computer centre personnel as well as computer users. The second adds a mechanism to improve the efficiency of a multiprogrammed computer system by potentially improving the mix of jobs entering the main processor at a point in time. These enhancements are not intended to correct deficiencies in the original formulation. Rather, they make use of the structure provided to account for more of the interactions arising between computer users and the computer centre.

(Received October 1974)

In an earlier article (Shaftell and Zmud, 1974) a flexible pricing system for allocating computer resources was described. Two enhancements to that control system are now presented: (*a*) a service quality dimension is incorporated within the flexible pricing system and (*b*) a mechanism is added to improve the efficiency of a computer system resulting in an increased throughput of jobs.

These additions are not intended to correct deficiences in the original formulation. Rather, they make use of the structure provided to account for more of the interactions arising between computer users and the computer centre.

Addition of a service quality dimension

A flexible pricing control system has as one of its aims increasing the effective capabilities of the existing computer facility. In the original formulation of the control system, a majority of this responsibility is placed upon the computer users. The computer user is induced through price fluctuations and a constrained budget to adopt usage patterns leading to efficient usage of the computer system. The computer centre is only indirectly affected by the control system: computer users are assumed to purchase services with which they are dissatisfied only reluctantly. By incorporating measures of service quality within the control system, the behaviour of computer centre personnel as well as computer users is influenced by the control system.

Although service quality is an elusive concept, information is available from which measures of service quality can be derived. The approach to be taken further develops an idea (Kirby and Raike, 1968) of using job turn-around time as a 'relative regret' function representing disappointment within a user community.

In the field study elsewhere discussed (Zmud, 1974), three factors emerged as being most important to computer users regarding their perceptions of service quality: job turnaround time, job re-running, and computer system availability. The first two factors are easily monitored; the third is not. It is difficult to ascertain whether a computer user actually needed access to the computer when he claims to have not gained access. Consequently, computer system availability is not considered in this discussion.

Job turnaround is unsatisfactory when, for example, a user requests a one hour turnaround and is willing to pay the price for one hour turnaround, but actually receives a much longer turnaround. A similar frustration arises when a user requests data be stored on a fast device but is given a slower device, or when a user requests the use of a high speed printer but receives a low speed printer. In general, it is possible to obtain a measure of service quality whenever multiple levels of a service are available to computer users. Dissatisfaction occurs when a poorer level of service than requested is received.

Job re-running, the other factor associated with service quality, measures a more obvious discontent—computer users are displeased when jobs must be reprocessed.

The measure of dissatisfaction for user i in control period j, d_{ij}, is the ratio of the cost of unsatisfactory service to the user divided by the gross cost of service to the user. Dissatisfaction costs are considered as the 'relative regret' of each user. The charge usually made against the user's budget is the net cost of service to the user—gross cost minus the cost of unsatisfactory service.

The measure of user dissatisfaction associated with resource k in control period j, d_{jk}, is the ratio of the cost of unsatisfactory responding to user requests for the resource divided by the gross service cost of the resource. For those resources not belonging to a substitutable class (i.e. multiple levels of a service), the cost of re-run jobs is the sole measure of service quality.

The measure of overall dissatisfaction in control periods j, d_j, is the ratio of the total cost for unsatisfactory service divided by the gross revenue of all services. As the cost of unsatisfactory service is not charged to the user community, it can be charged to the computer centre and used in performance evaluation.

The service quality measures defined above are stated in the notation and terminology of the original control model in Appendix 1.

Two gains result from adding a service quality dimension to the flexible pricing control system. Computer users are not charged for unsatisfactory services, and the computer centre is explicitly evaluated with respect to the quality of service being provided. Service quality measures can be included in the exception reports and resource audits associated with the original control system.

In addition, the second step of the solution procedure (estimation of the price-demand relationships) is enhanced by the addition of a service quality variable to the demand function. Using the demand function employed in the example problem, the modified demand function would appear as:

$$\sum_{i \in I} x_{ijk} = C_k \, d_{jk}^{xk} \prod_{\substack{q \in Q \\ \text{s.t.} e \\ (k \in Q_l)}} p_{jq}^{x_{kq}}$$

The demand for a resource is now a function of the price of the resource, the prices of substitutable resources, and the user dissatisfaction, d_{jk}, attributed to the resource. In estimating resource prices for the current control period, the last period's dissatisfaction measures (or a weighted average of recent dissatisfaction measures) are used.

Increasing computer system throughput

Modern computer systems generally employ 'shared-resource' computing environments such as multiprogramming, time-sharing, and multiprocessing. Each of these environments finds multiple jobs enjoying concurrent access to the same set of resources.

There are three states a job may assume once it gains access to the computer system: active, ready, or wait. An active job currently is using the central processing unit (cpu). There can be as many active jobs as there are cpu's. (A multiprocessing system employs multiple cpu's.) A job is in a ready state if it is ready to use the cpu but cannot because the cpu is busy serving another job. A job waiting for the completion of some activity (input/output, operator, etc.) and unable to use the cpu is in a wait state. A well-balanced situation will find the computer system filled with sufficient jobs so that the probability is high some job will always be in a ready state.

Computer job requirements are classified along a continuum from being cpu-bound to being I/O-bound. A cpu-bound job possesses a heavy demand for the cpu and little demand for input/output activity. An I/O-bound job is just the opposite.

The efficiency of a 'shared-resource' computer system is dependent on the job mix. A mix containing an I/O-bound job with a cpu-bound job is more desirable than a mix of two jobs of the same classification. If the mix is perfect (that is, a job is always ready to use the cpu when it becomes available), 100 per cent utilisation and maximum job throughput is achieved.

In addition to a job's cpu and I/O requirements, a job's requirements for other resources may also affect its suitability to be processed with other jobs. If an excessive amount of a resource (such as core space) is required by a job, the number of other jobs with which it can be efficiently mixed is reduced. The probability of attaining a high degree of cpu usage is correspondingly reduced.

As computer system efficiency is dependent upon obtaining suitable mixes of jobs for concurrent processing, it is desirable that a job's requirements be known prior to the job entering the computer system. When job requirements are available, pre-scheduling algorithms can be used to arrive at an efficient mixing of those jobs desiring service at a given point in time.

Job requirements can be made available in two ways. First, data files can be maintained containing past histories of job requirements. When a job enters the pre-scheduler, this file is searched and the appropriate information is withdrawn. This approach has drawbacks. It requires large amounts of data storage and analysis, and it provides no estimates of job requirements for the initial running of a job. Second, the computer user can provide estimates of job requirements when submitting jobs to be processed. Assuming users are motivated to provide accurate estimates, this second approach is preferred as fewer organisational resources are needed.

The motivation for users to furnish accurate estimates can be provided (McKell and Moskowitz, 1972) through a pricing system by rewarding those users who contribute to efficiency and penalising those who do not. One such scheme has been implemented at the University of Rochester (Swoyer and Armstrong, 1969) and has resulted in a marked increase in throughput.

In situations where a decision maker prepares forecasts of his expected resource usage, pressures for misestimation are high (Ijiri, Kinard, and Putney, 1968). As the decision maker is evaluated on the degree actual usage agrees with forecasted usage, the estimates provided often allow for uncertainty or error on the part of the decision maker.

Similar pressures affect the computer user required to estimate resource requirements. It takes time and effort for the user to make the estimates. If he perceives other activities to be of greater importance, the effort spent preparing estimates will be minimal. In addition, operating system characteristics may lead the user to give false estimates. Scheduling algorithms may induce users to understate their needs. For example, jobs requiring less than 100,000 words of core space may be given priority over jobs not meeting this condition. To gain faster turnaround, core requirements may be falsely specified. Processing defaults may induce users to overstate their needs. For example, jobs exceeding stated requirements may be terminated prior to completion. If a user is unaware of his true need, a larger than necessary safety margin may be included with the estimate.

An addition to the flexible pricing system earlier presented can provide a proper motivating force. The following notation is used for the general case:

C = actual charge
C' = initial charge
E = estimated usage
A = actual usage
s = penalty coefficient ($s \geq 0$).

Then, if it is desired to meet the following conditions,

$$C = C' \text{ when } A \neq E;\ C = C' \text{ when } A = E\ ,$$

the charging algorithm becomes:

$$C = C' + s|E - A|$$

When actual usage equals estimated usage, the charge is unchanged. However, when actual usage differs from estimated usage, the charge increases as a penalty for misestimation.

Often, it is not critical that actual usage exactly equal estimated usage. In addition, it may be desired to reward those individuals who provide accurate estimates. The general case is easily modified to represent this situation. Let σ equal the percentage of mis-estimation allowed and restate the conditions as:

$$C = C' \text{ when } \frac{|E - A|}{E} = \sigma$$

$$C > C' \text{ when } \frac{|E - A|}{E} > \sigma$$

$$C < C' \text{ when } \frac{|E - A|}{E} < \sigma$$

Then, the charging algorithm becomes:

$$C = C' + s\left(\frac{|E - A|}{E} - \sigma\right)$$

If it is desired that rewards be felt different from penalties, the penalty coefficient can take on different values depending on whether the actual deviation is greater or less than the allowed deviation.

The above formulation is expressed in the notation and terminology of the original control model in Appendix 2.

Two problems have not been addressed: selecting those resources needing requirement estimates and determining the rates to use as penalty coefficients. The selection of those resources requiring estimates is dependent on the operating system and scheduling algorithm being used. Hence, the set of resources will vary with each computer installation. The determination of optimal penalty coefficients is a more difficult problem. These rates can be heuristically derived via a trial and error procedure. In addition, research on analytic derivations of penalty coefficients is reported in the literature (McKell and Moskowitz, 1972).

Appendix 1 Addition of a service quality measure to the flexible pricing control system

The notation of the original model is retained and some new notation is added. Let

I = (all users)
K = (all resources)
K^* = (all 'multi-levelled' resources, $K^* \in K$)
x_{ijk} = amount of resource k used by user i in control period j
x^*_{ijk} = amount of resource k used by user i in control period j where $k \in K^*$ and the level of service received is unsatisfactory (other than a job re-run)
x^{**}_{ijk} = amount of resource k used by user i in control period j where the job had to be re-run
p_{jk} = price of a unit of resource k during control period j; if $k \in K^*$ the price is the minimum of the prices of the service level requested and the service level received
p^*_{jk} = price of a unit of resource k during control period j where $k \in K^*$ and the level of service is unsatisfactory; resource k refers to the requested level of service
d_{ij} = measure of dissatisfaction for user i in control period j
d_{jk} = measure of dissatisfaction for resource k in control period j
d_j = overall measure of dissatisfaction in control period j

then,

$$d_{ij} = \frac{\sum_{k \in K^*} (p^*_{jk} - p_{jk}) x^*_{ijk} + \sum_{k \in K} p_{jk} x^{**}_{ijk}}{\sum_{k \in K} p_{jk} x_{ijk}}$$

$$d_{jk} = \frac{\sum_{i \in I} (p^*_{jk} - p_{jk}) x^*_{ijk} + \sum_{i \in I} p_{jk} x^{**}_{ijk}}{\sum_{i \in I} p_{jk} x_{ijk}}$$

$$d_j = \frac{\sum_{i \in I} \sum_{k \in K^*} (p^*_{jk} - p_{jk}) x^*_{ijk} + \sum_{i \in I} \sum_{k \in K} p_{jk} x^{**}_{ijk}}{\sum_{i \in I} \sum_{k \in K} p_{jk} x_{ijk}}$$

Appendix 2 Addition of a mechanism for increasing throughput to the flexible pricing control system

The notation of the original model is retained and some new notation is added. Let

K = (all resources)
K' = (all resources needing estimates, $K' \in K$)
s_{jk} = penalty coefficient for resource k in control period j ($s_{jk} \geq 0$)
σ_{jk} = percentage deviation allowed for resource k in control period j
p_{jk} = price of a unit of resource k during control period j
e_{ijk} = estimated usage of resource k by user i for a single job in control period j
a_{ijk} = actual usage of resource k by user i for a single job in control period j.

Then, the actual charge assessed user i for the resources used in a job during control period j is:

$$\sum_{k \notin K'} p_{jk} a_{ijk} + \sum_{k \in K'} s_{jk} \left(\frac{|e_{ijk} - a_{ijk}|}{e_{ijk}} - \sigma_{jk} \right) p_{jk} a_{ijk}$$

References

IJIRI, Y., KINARD, J. C., and PUTNEY, F. B. (1968). An Integrated Evaluation System for Budget Forecasting and Operating Performance with a Classified Budgeting Bibliography, *Journal of Accounting Research*, Vol. 6, No. 1, pp. 1-11.

KIRBY, M. J. L., and RAIKE, W. M. (1968). Priority Planning in a University Computation Centre, *CBA Working Paper* 69-14, University of Texas at Austin.

MCKELL, LYNN J., and MOSKOWITZ, HERBERT (1972). An Integrated Planning and Control Model for a Close Systems Environment with Application to Computer Processing Systems, Proceedings, *Fourth Annual Meeting of AIDS*, pp. 43-50.

SHAFTELL, T. L., and ZMUD, R. W. (1974). Allocation of Computer Resources through Flexible Pricing, *The Computer Journal*, Vol. 17, No. 4, pp. 306-312.

SWOYER, VINCENT H., and ARMSTRONG, MICHAEL F. (1969). Timing and Charging Computer Use on the IBM 360 Model 65 in a Multiprogrammed Environment, *Technical Report* 21-69, University of Rochester, New York.

ZMUD, R. W. (1974). Towards an Understanding of the Computer Center/User Interface, unpublished Ph.D. dissertation, University of Arizona, Tucson.