Pergamon

0360-8352(94)E0017-G

Computers ind. Engng Vol. 26, No. 4, pp. 743–756, 1994

0360-8352/94 $7.00 + 0.00

OPTIMIZATION MODELS FOR COMPUTER DATA STORAGE DESIGN: AN APPLICATION

MARCO GAGLIARDI[1] and COSIMO SPERA[2,*]

[1]Department of Statistic, Probability and Applied Statistic, University of Rome "La Sapienza", p.zzale A. Moro, 00185 Rome, Italy and [2]Department of Industrial Engineering and Operation Research, University of Michigan, 1205 Beal Av., Ann Arbor, MI 48109, U.S.A.

(Received for publication 7 April 1994)

Abstract—In this paper we discuss a model being used to optimize the system design of the Computer Centre of one of the most important Italian banking groups. Data and transactions, processed by the system, are grouped respectively in data sets and by type, so it is possible to deal with the large dimensions of the corresponding optimization models. The transactions' arrivals are considered as stochastic variables and their probability values are estimated on the base of theoretical considerations. The solutions for two optimization problems, constructed and solved for different scenarios, are discussed in detail.

1. THE DESCRIPTION OF THE PROBLEM

The scenario described by our model considers the daily transactions arriving at the main computer centre of an important Italian banking group. All the computerized banking operations are managed by this centre.

For a better understanding a picture of our scenario is given in Fig. 1. There are four basic entities: (a) the data storage units (D.S.U.); (b) the data requested by the transactions; (c) the time periods; and (d) the typology of the transactions; their descriptions follow:

(a) The computer centre uses a variety of DSUs to store the data. These units can be grouped according to their performances on memory levels, and ordered with respect to access time and transfer rate. There are four memory levels: (1) advanced disks, (2) disk, (3) packed data disks and (4) cartridges.

The distinction between advanced disks and disks, is mainly due to the fact that the former were purchased at the time when at higher technology corresponded a much lower price, so they have faster access time and transfer rate and a lower cost function. Disks are kept, because financially they have to be amortized yet. Packed data disks and cartridges memorize compressed data, which to be used, have to be decompressed. Handle Robots search among the cartridges, put the selected ones in the input/output units and when the requests are completed, put them back. Average search time is added in the decompression time.

Every memory level is characterized by a maximum available space, storage cost, access time, decompression time and transfer rate, the actual data values are shown in Table 1.

(b) Generally a bank deals with a very large amount of information. Data concern the checking and saving accounts of the customers, loans, stock market transactions, etc. By similarity, data can be grouped together in data sets, which are stored on different memory levels. We consider the proportions of data sets stored on different memory levels as variables and their values are obtained by solving the optimization problems. One of the reasons to work with proportions is that numerical roundoff errors can be avoided. Table 2 shows the dimensions expressed in Gbytes of the 10 data set considered.

(c) We consider the day as the time period. This is split into three different homogeneous periods. The homogeneity is considered with respect to the behaviour of the transactions arriving at the centre. The three periods are: (1) 0–8 a.m., (2) 8 a.m.–6 p.m. and (3) 6–12 p.m. The second period corresponds to the office hours.

*On leave from University of Siena, Italy on a CNR fellowship. To whom all correspondence should be addressed.

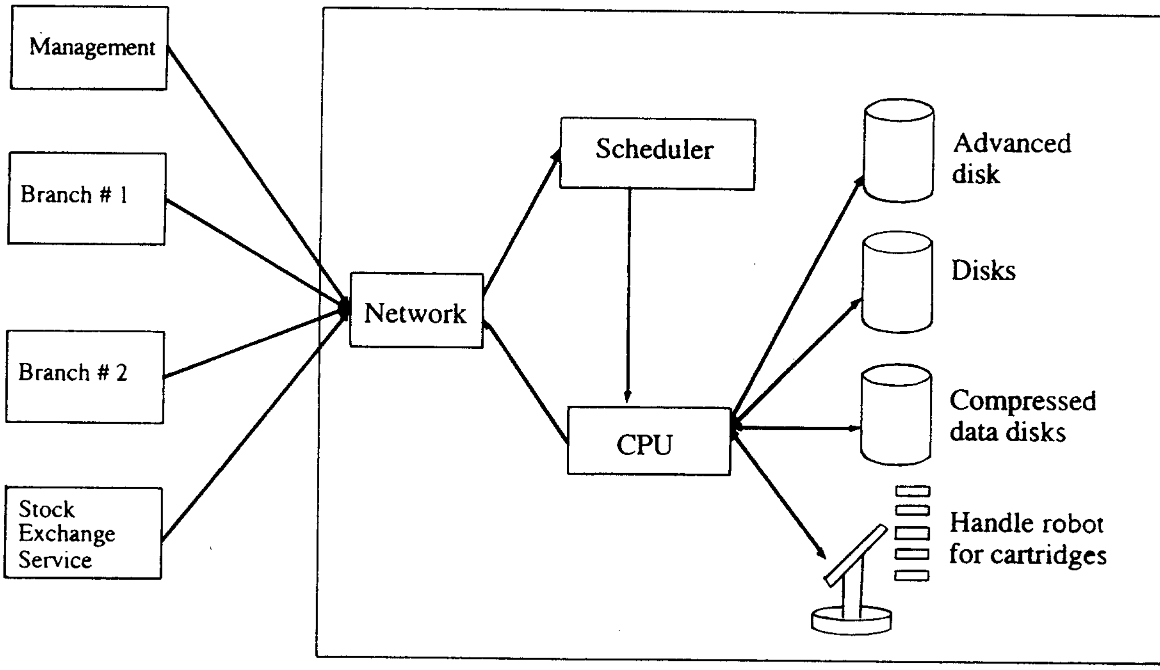

Fig. 1. Scenario representation.

(d) Once a transaction arrives, it has to be completed in the pre-fixed amount of time. The transaction uses the data sets stored in certain memory levels as resources, produces new information and eventually updates the data sets. Transactions are grouped by typology, each one is characterized by the maximum completion time, the data sets and memory levels used. Table 3 provides the actual values for our model.

Tele-processed transactions (TP) are requested by the users at the bank's branches and processed by the central computer centre. There are two kinds of teleprocessed transactions: (1) customers' TP; and (2) management's TP. The first concern the services offered to the customers, the second are transaction requested by the management of the bank. At the end of the working day these TP transactions need to be revised. The post TP transaction groups them and executes this revision. In Table 3, other three typologies appear: "bancomat" is the transaction typology for the cash withdrawal operations; "batch" is the transaction executed in batch mode; "security" is the transaction to make back-up copies of the files. Also reported are: the pre-fixed completion time expressed in seconds; the used data sets t-uple, which indicates the data sets used by the transaction; the used memory level t-uple which indicates the memory level used by this transaction for the current situation, i.e. before running any of the optimization problems.

The decision makers can follow two possible strategies:

(i) minimize the global storage cost;
(ii) maximize the performances of the system, which implies minimizing the transactions' completion time.

The global storage cost is a function of the data set distributions among the storage levels. The associated cost to each variable is computed for each memory level as the weighted average among maintenance, amortized and space renting costs.

Table 1. Data storage units attributes; storage cost is expressed in Italian lire

Id	DSU	Available space (Gbytes)	Storage cost	Access time	Decompression	Transfer rate (mbyte/sec)
1	Advanced Disk	324.88	607,000	6–15 msec	Null	10.5
2	Disk	191.1	739,200	30 msec	Null	4.5
3	Packed data Disk	135.9	468,800	30 msec	1–4 min	4.5
4	Cartridge	1133.4	4585	20–40 sec	2–7 min	4.5

Table 2. Data sets' attributes

Id	Data set	Dimension (Gbytes)
a	Checking accounts	202.5
b	Stocks and Bonds	69.8
c	Registry data	21.4
d	Bank book-keeping	86.5
e	Transfer orders	17.3
f	Portfolio	28.6
g	Securities, Saving accounts	26.9
h	Central Bank control data	168.4
i	External banks data	20.4
j	Other data	781.4

Table 3. Transactions' attributes

Transaction typology	Prefixed completion time (sec)	Used data set	Used memory level
Customers' TP	0.281549	a, b, c, d, e, g, h, j	1
Management's TP	0.629801	All data sets	1, 2
Post TP	1297.601	All data sets	1, 2
Batch	0.629801	All data sets	2, 3, 4
Bancomat	0.209504	a, b, c, e, j	1
Security	1297.3	All data sets	3, 4

It is a reasonable hypothesis to define a linear relation between the proportion of data sets and the memory levels where these data sets are stored, therefore the model corresponding to the first strategy has a linear objective function.

System performances depend mainly on two sets of attributes: data set storage levels and transaction arrivals' distributions. These distributions can either be obtained analysing the data collected by the centre or formulating some distribution hypotheses on the basis of theoretical considerations.

2. MATHEMATICAL FORMULATION

In this section we give the mathematical formulation of our model. We want to point out, as shown in [1], that the decision makers have formulated their optimization problems using an objected oriented language called BLOOMS.

This language satisfies the core concepts of the structured modeling, as defined by Geoffrion [5]. However it is part of a Model Management System under development by the authors at University of Siena. A complete detailed description of BLOOMS and its grammar is given in [2, 4].

Table 4 gives the description of the parameters and variable used.

There are four sets of constraints to consider:

2.1. Data sets are distributed over the memory levels. The sum of the proportions for each data set must be equal to one. This set of constraints is called **proportion constraints** and has cardinality equal to I.

$$\sum_{j=1}^{J} x_{ij} = 1 \quad i = 1, \ldots, I. \tag{1}$$

Table 4. Parameters and variable

	Index range	Description		Index range	Description
a_j	$j = 1, \ldots, J$	jth memory level access time	x_{ij}	$i = 1, \ldots, I$; $j = 1, \ldots, J$	Proportion of the ith data set stored in the jth memory level
t_j	$j = 1, \ldots, J$	jth memory level transfer rate	u_{ik}	$i = 1, \ldots, I$; $k = 1, \ldots, K$	ith data set used by kth transaction typology; it is a binary value: 0 if the data set is not used, 1 otherwise
b_j	$j = 1, \ldots, J$	jth memory level dimension	p_{jk}	$j = 1, \ldots, J$; $k = 1, \ldots, K$	jth memory level used by the kth transaction typology: it is a binary value
c_j	$j = 1, \ldots, J$	jth memory level cost.	f_{kh}	$k = 1, \ldots, K$; $h = 1, \ldots, H$	Arrivals' probability values for the kth transaction. These values are defined for each homogeneous period h
d_i	$i = 1, \ldots, I$	ith data set dimension	m_k	$k = 1, \ldots, K$	Pre-fixed completion time for the kth transaction typology

In our application the values for J and I are respectively 4 and 10, therefore the model has 40 variables and 10 proportion constraints.

2.2. Proportions of data set stored in a memory level cannot exceed the storage space available. Therefore:

$$\sum_{i=1}^{I} d_i x_{ij} \leqslant b_j \quad j = 1, \ldots, J. \tag{2}$$

This set is called **space constraints**; its cardinality is equal to J. Obviously there are 4 space constraints.

2.3. For every type of transaction there is a completion time. This can not exceed a given pre-fixed time value, which is its upper bound and it is decided by the management, so in the model it is a given constant. For each memory level set:

$$\text{TRANSF}_j = a_j + t_j^* \quad j = 1, \ldots, J \tag{3a}$$

t_j^* indicates the time expressed in seconds needed to transfer a Mbyte data file stored on the jth memory level into the memory, a_j being the access time to the jth memory level. Therefore TRANSF_j indicates the total time needed to access and use this data file.

Let T_k be the kth transaction typology completion time. It depends on the TRANSF_j values and it is computed as the sum of the execution time T_k' plus a penalty time Q_k. T_k' is the product between the TRANSF_j values and the proportion of data set stored in the jth memory level; the penalty time Q_k is computed as the sum for each utilized data set and for each not utilized memory level of the product between the TRANSF_j value and the squared proportion of data set plus one. It has the following meaning: it is the price to pay when a transaction uses memory levels not in its t-uple as indicated in Table 3. It has been defined as a quadratic function of x_{ij}.

The reason to have the penalty defined as quadratic function is to compensate the decreasing value of T_k' which occurs when typologies use faster memory levels not in their t-uple.

$$T_k' = \sum_{i=1}^{I} u_{ik} \sum_{j=1}^{J} p_{jk} x_{ij} \text{TRANSF}_j$$

$$Q_k = \sum_{i=1}^{I} u_{ik} \sum_{j=1}^{J} (1 - p_{jk})(1 + x_{ij})^2 \text{TRANSF}_j$$

$$T_k = T_k' + Q_k. \tag{3b}$$

If Q_k^* is the penalty value for the current data set proportions, then the right-hand side m_k^* is as:

$$m_k^* = Q_k^* + m_k$$

where m_k is the pre-fixed completion time. Therefore, we define the set of constraints as:

$$T_k \leqslant m_k^* \quad k = 1, \ldots, K. \tag{3c}$$

This set of constraints is called **time constraints**; its cardinality is equal to K. There are 6 time constraints in our model.

2.4. Two or more transactions can be requested to be completed at the same time. In this case conflicts in the management of the resources can arise, which make the set of time constraints unfeasible. Conflicts are only possible among different transaction typologies, since same transaction typologies use the resources in a compatible way to be completed in parallel. Transactions' arrivals are not deterministic; therefore, we define a set of constraints which consider these probabilistic values and try to avoid unfeasibility.

Given the time period h and the transaction typology k, f_{kh} indicates the probability value that in a pre-defined interval a transaction of the type k arrives. If we assume that the arrivals' probability distributions are independent, we can compute the conflict probability α_{sh} for every transaction combination $s \in S$, where S is the power set constructed on the set of the transactions, at period h:

$$\alpha_{sh} = \prod_{k \in s} f_{kh} \quad \forall\, s \in S, \quad h = 1, \ldots, H. \tag{4a}$$

Let us define a function which returns 0 if the probability of conflicts is equal to 0, 1 otherwise.

$$\delta(\alpha_{sh}) = \begin{cases} 0 & \alpha_{sh} = 0 \\ 1 & \text{otherwise} \end{cases}.$$

Computed the α_{sh} and δ_{sh} values we can construct the set of constraints as follows:

$$\delta_{sh} \sum_{k \in s} T_k \leqslant (1 - \alpha_{sh}) \sum_{k \in s} m_k^* + \alpha_{sh} \max_{k \in s} m_k^* \quad \forall\, s \in S, \quad h = 1, \ldots, H. \tag{4b}$$

The left-hand side of the inequalities is the sum of the transaction completion times when transactions have to be completed sequentially, multiplied by δ_{sh} values. The constraints are always satisfied when the conflict probability of the set of transaction is equal to 0. The right-hand side is a function of α_{sh} values. Let us point out that this set of constraints includes the time constraint set (3c). In fact, when $s = \{k\}$, $k = 1, \ldots, K$, then inequality (4b) becomes:

$$T_k \leqslant m_k^* \quad k = 1, \ldots, K$$

which corresponds to (3c). For this reason we redefine the set S excluding the singletons. Let G be the set of all the transaction typologies and $P(G)$ the power set construct on G then:

$$S = \{P(G)/\varnothing\}/\{k\} \quad k = 1, \ldots, K.$$

The set of constraints (4b) is called **distribution constraints.** It has cardinality equal to $(2^K - (K+1)) * H$.

Given that the distribution constraints have to be computed for the different homogeneous time periods there are $57 \times 3 = 171$ constraints.

Based on the previous two strategies the following two optimization problems can be considered.

Model I: minimize global storage cost

$$\min \sum_{j=1}^{J} c_j \sum_{i=1}^{I} d_i x_{ij}$$

s.t.

$$\sum_{j=1}^{J} x_{ij} = 1 \quad i = 1, \ldots, I \qquad \delta_{sh} \sum_{k \in s} T_k \leqslant (1 - \alpha_{sh}) \sum_{k \in s} m_k^* + \alpha_{sh} \max_{k \in s} m_k^* \; \forall\, s \in S, \quad h = 1, \ldots, H$$

$$0 \leqslant xij \leqslant 1 \quad i = 1, \ldots, I; \quad j = 1, \ldots, J$$

$$\sum_{i=1}^{I} d_i x_{ij} \leqslant b_j \quad j = 1, \ldots, J$$

$$T_k \leqslant m_k^* \quad k = 1, \ldots, K$$

Goal II: maximize system performances

$$\min \sum_{k=1}^{K} \left[\sum_{i=1}^{I} u_{ik} \sum_{j=1}^{J} p_{jk} x_{ij} \mathrm{TRANSF}_j + \sum_{i=1}^{I} u_{ik} \sum_{j=1}^{J} (1 - p_{jk})(1 + x_{ij})^2 \mathrm{TRANSF}_j \right]$$

s.t.

$$\sum_{j=1}^{J} x_{ij} = 1, \quad i = 1, \ldots, I \qquad \delta_{sh} \sum_{k \in s} T_k \leqslant (1 - \alpha_{sh}) \sum_{k \in s} m_k^* + \alpha_{sh} \max_{k \in s} m_k^* \; \forall\, s \in S, \quad h = 1, \ldots, H$$

$$\sum_{i=1}^{I} d_i x_{ij} \leqslant b_j \quad j = 1, \ldots, J \qquad \sum_{j=1}^{J} c_j \sum_{i=1}^{I} d_i x_{ij} \leqslant \text{UPPER\_BOUND\_COST}$$

$$0 \leqslant x_{ij} \leqslant 1, \quad i = 1, \ldots, I; \quad j = 1, \ldots, J.$$

The model has 40 variables and a total of 191 constraints. When the storage cost is minimized and 40 variables and 185 constraints when the system performance is maximized.

Both are non linear models, which have been solved using the Robinson algorithm [10] coded in Minos 5.0 software package [7].

3. SIMULATION SCENARIOS

In order to give an example, we need to set the probability distribution values f_{kh} for the transactions' arrivals in three considered homogeneous periods. Presently, the Computer Centre monitors only the TP transactions, so their frequency values are available for each hour of the considered period. There is no record for the other transactions.

It has been noted that for these monitored transactions the observed data follow a stochastic Poisson process. So, we decided to assume also for the other non monitored transaction that the transactions arrive as a stochastic Poisson process $K(t)$, $t \geqslant 0$. For any time interval of length τ, the number of transactions requested by the users follow the Poisson distribution with rate λ.

Therefore, for each transaction k, given the time t in the homogeneous period h, the probability that the number of requested transactions if equal to n in the interval τ is:

$$P\{K(t+\tau)-K(t)=n\}=\frac{(\lambda_{kh}\tau)^n}{n!}\exp(-\lambda_{kh}\tau). \quad n=0,1,\ldots$$

This value depends on λ_{kh}, which can vary for different transactions and periods. The Poisson stochastic process has an interesting property: the inter arrivals are independent and exponentially distributed.

Let us consider a transaction typology k and an arbitrary time h_0 in h; at h_0 a transaction has arrived. If we indicate with h_k the time when the next transaction arrives; the probability, computed when the time is h_0, that the inter arrival period $\Delta_k = h_k - h_0$ is less or equal to a pre-defined interval $\Delta^* = h^* - h_0$ is:

$$P\{\Delta_k \leqslant \Delta^*\} = 1 - \exp(-\lambda_{kh}\Delta^*).$$

This probability value is function of the λ_{kh} rate and of the length of Δ^*. Since the period h is considered to be homogeneous, then the probability value is the same for intervals of the same length. This situation is illustrated in Fig. 2.

To compute the f_{kh} values needed to instance the model we must specify the values for the K^*H rates λ_{kh} and the pre-defined time interval Δ^*. Its length is the minimum among the pre-fixed completion times of transaction typologies. This implies that:

(a) a sub-set of transaction $s \in S$ has a compound arrival probability in Δ^* equal to:

$$P\{\Delta_k \leqslant \Delta^*, k \in s\} = \prod_{k \in s} P\{\Delta_k \leqslant \Delta^*\} = \prod_{k \in s} [1 - \exp(-\lambda_{kh}\Delta^*)] \quad \forall s \in S, \quad h = 1, \ldots, H.$$

(b) the conflict probability (2.4a) for the transactions is derived from the above formula. In fact the f_{kh} values are computed as follows:

$$f_{kh} = P\{\Delta_k \leqslant \Delta^*\} = 1 - \exp(-\lambda_{kh}\Delta^*) \quad k = 1, \ldots, K, \quad h = 1, \ldots, H.$$

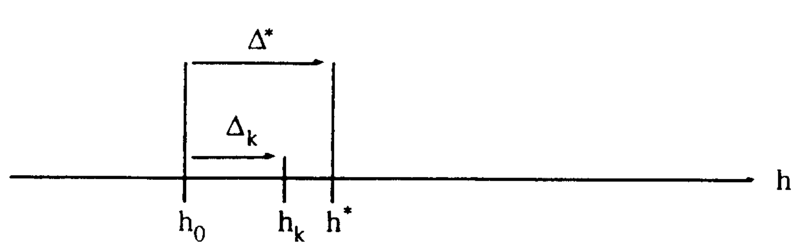

Fig. 2.

The interval Δ^* is normalized, so as to be considered as unitary. The arrival rates λ_{kh} have the meaning of "means of arrivals of the kth transaction within the unitary period, at time period h", and the f_{kh} values mean "probability that next request of kth transaction typology arrives within the unitary interval".

The two optimization problems can be solved for different values of the parameters. Any combination of these values defines a different scenario. Here we have listed the scenarios that our decision makers have interest to solve:

3.1. Different transaction typologies with the same or different arrival rates

The simplest hypothesis is to set, for the homogeneous periods considered, the values of the arrival rates equal for all the transactions; unless some particular transaction is not required. It is shown in Table 5 as the transactions' distribution over the day. Here if the value is equal to zero, it means that there are no arrivals for that transaction in that period and if it is equal to one that transaction arrives at a particular rate. This is fixed to be any value among: $\{0.2, 0.5, 1.0\}$. These values have been selected with the following motivations:

(a) The arrival rate 1.0 implies that in the considered unit time the transaction arrives on average once. This arrival rate value creates many conflicts as it is shown from the computational results presented in the next section.

(b) The arrival rate 0.2 is the actual value computed from the available data for the monitored transactions.

(c) The arrival rate 0.5 simulates a reasonable increment of the daily transactions.

Note that if the probability conflict values f_{kh} are equal to zero, ($\lambda_{kh} = 0$), then the distribution constraints are always satisfied.

3.2. Modifications of the memory space for the DSUs

Here two different strategies are analyzed:

3.2.1.: Memory levels are increased uniformly by 10%. This implies that now the right-hand side for the space constraints becomes:

$$b_j = b_j * 1.1 \quad j = 1, \ldots, J.$$

3.2.2.: Disks are replaced before they are amortized and this memory level is upgraded to advanced disks. Now the right-hand side for the space constraints is:

$$b_1 = b_1 + \text{old\_}b_2$$

$$\text{new\_}b_2 = 0.$$

3.3. Budget limit

The current storage cost for the bank is 357 million Italian lire per month. In the second optimization problem, where the performance of the system is maximized, we try to bring this value to 310 and 257 million. The first is the storage cost which can be easily achieved making simple variation at the current data set distribution, the second is the budget goal for our decision makers.

Table 5. Transactions' distribution

Transaction typology	h_1	h_2	h_3
Customers' TP	0	1	0
Management's TP	0	1	0
Post TP	1	0	1
Batch	1	1	1
Bancomat	1	1	1
Security	1	1	1

Table 6

	$\lambda = 0$				$\lambda = 0.2$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.22	0	0	0.78	0.268	0	0	0.732
Registry data	0.8	0.2	0	0	0.8	0.2	0	0
Bank book-keeping	0.2	0	0	0.8	0.2	0	0	0.8
Transfer orders	0.8	0.2	0	0	0.8	0.2	0	0
Portfolio	0.2	0	0	0.8	0.23	0	0	0.77
Securities	0.8	0.152	0	0.048	0.8	0.2	0	0
Central Bank control data	0.2	0	0	0.8	0.2	0	0	0.8
External bank data	0.701	0	0	0.299	0.737	0	0	0.263
Others	0.026	0	0.174	0.8	0.026	0	0.174	0.8
	$\lambda = 0.5$				$\lambda = 1$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.521	0	0	0.479	0.8	0.17	0	0.03
Registry data	0.8	0.2	0	0	0.8	0.2	0	0
Bank book-keeping	0.2	0	0	0.8	0.507	0	0	0.493
Transfer orders	0.8	0.2	0	0	0.8	0.2	0	0
Portfolio	0.522	0	0	0.478	0.8	0.01	0	0.19
Securities	0.8	0.2	0	0	0.8	0.2	0	0
Central Bank control data	0.2	0	0	0.8	0.192	0.008	0	0.8
External bank data	0.8	0	0	0.2	0.8	0.2	0	0
Others	0.026	0	0.174	0.8	0	0.026	0.174	0.8

Table 7

	$\lambda = 0$				$\lambda = 0.2$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.22	0	0	0.775	0.274	0	0	0.726
Registry data	0.8	0.2	0	0	0.8	0.2	0	0
Bank book-keeping	0.2	0	0	0.8	0.2	0	0	0.8
Transfer orders	0.8	0.2	0	0	0.8	0.2	0	0
Portfolio	0.2	0	0	0.8	0.230	0	0	0.770
Securities	0.8	0.151	0	0.049	0.8	0.2	0	0
Central Bank control data	0.2	0	0	0.8	0.2	0	0	0.8
External bank data	0.7	0	0	0.3	0.737	0	0	0.263
Others	0.009	0	0.191	0.8	0.009	0	0.191	0.8
	$\lambda = 0.5$				$\lambda = 1$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.509	0	0	0.491	0.8	0.110	0	0.09
Registry data	0.8	0.2	0	0	0.8	0.2	0	0
Bank book-keeping	0.23	0	0	0.74	0.608	0	0	0.392
Transfer orders	0.8	0.2	0	0	0.8	0.2	0	0
Portfolio	0.473	0	0	0.527	0.759	0	0	0.241
Securities	0.8	0.2	0	0	0	0.8	0.2	0
Central Bank control data	0.2	0	0	0.8	0.2	0	0	0.8
External bank data	0.8	0.013	0	0.187	0.8	0.120	0	0.08
Others	0.009	0	0.191	0.8	0.009	0	0.191	0.8

Table 8

	$\lambda = 0$				$\lambda = 0.2$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.426	0	0	0.574	0.466	0	0	0.534
Registry data	0.8	0	0.16	0.04	0.8	0	0.169	0.031
Bank book-keeping	0.2	0	0	0.8	0.2	0	0	0.8
Transfer orders	0.8	0	0.2	0	0.8	0	0.2	0
Portfolio	0.2	0	0	0.8	0.305	0	0	0.695
Securities	0.8	0	0.08	0.12	0.8	0	0.033	0.167
Central Bank control data	0.2	0	0	0.8	0.2	0	0	0.8
External bank data	0.701	0	0	0.299	0.791	0	0	0.209
Others	0.038	0	0.162	0.8	0.036	0	0.164	0.8
	$\lambda = 0.5$				$\lambda = 1$			
Checking accounts	0.5	0	0	0.5	0.588	0	0	0.412
Stocks and bonds	0.724	0	0	0.276	0.8	0	0.180	0.02
Registry data	0.8	0	0.2	0	0.8	0	0.2	0
Bank book-keeping	0.202	0	0	0.798	0.8	0	0	0.2
Transfer orders	0.8	0	0.2	0	0.8	0	0.2	0
Portfolio	0.620	0	0	0.38	0.735	0	0	0.265
Securities	0.8	0	0.113	0.087	0.8	0	0.2	0
Central Bank control data	0.2	0	0	0.8	0.215	0	0	0.785
External bank data	0.8	0	0	0.2	0.8	0	0	0.2
Others	0.4	0	0.160	0.8	0.059	0	0.141	0.8

Table 9. Objective function value × 1000 Italian lire

Current situation		Uniformly increased 10%		Second level replaced	
Lambda	Objective function	Lambda	Objective function	Lambda	Objective function
0	235,1852	0	233,5473	0	240,6344
0.2	239,127	0.2	237,4963	0.2	244,5662
0.5	255,5859	0.5	254,1300	0.5	262,7703
1	302,8971	1	298,2385	1	320,3277

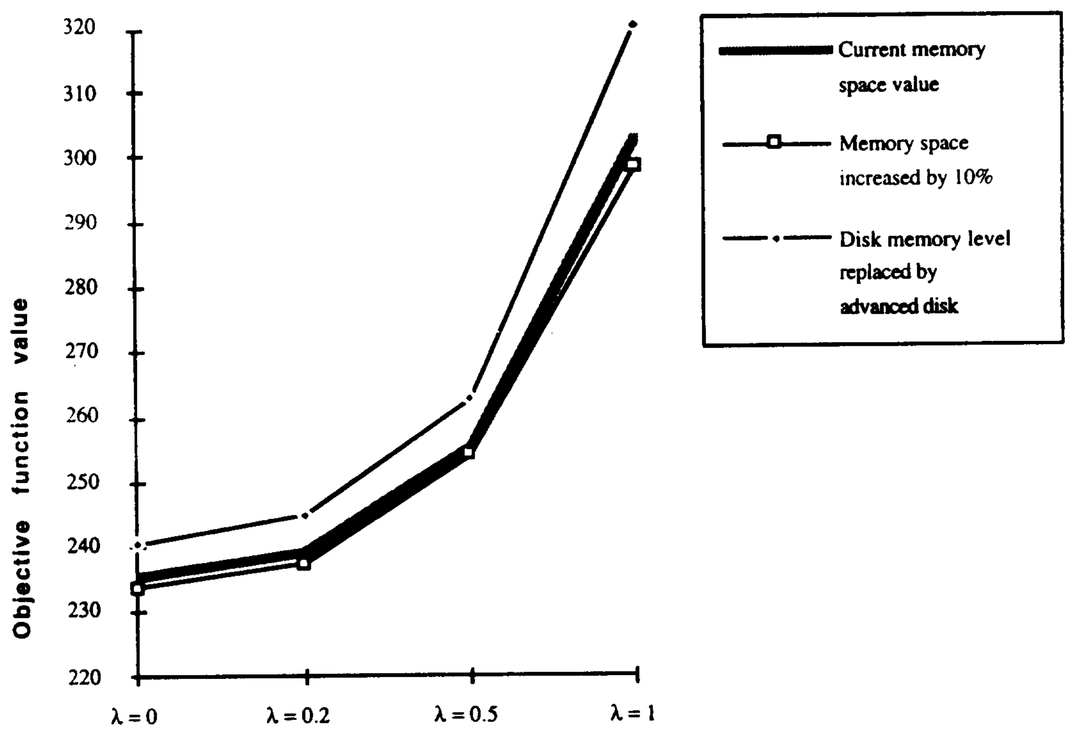

Fig. 3. Objective functions' values.

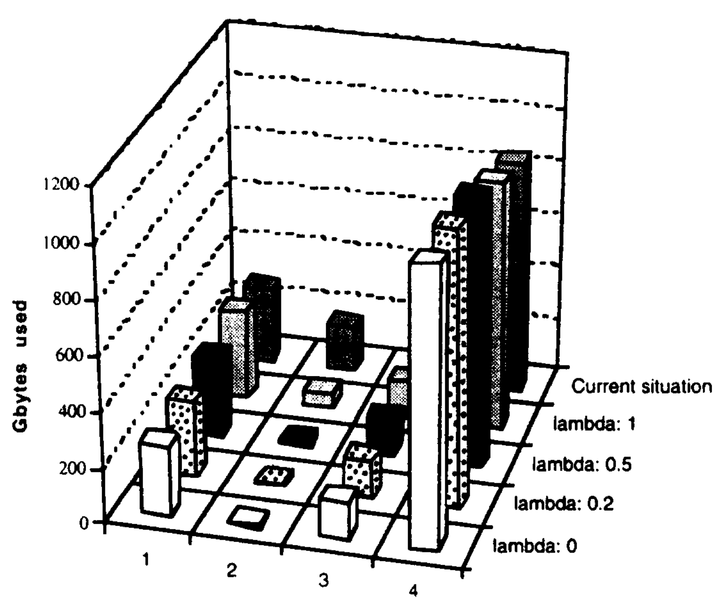

Fig. 4. Current memory level values.

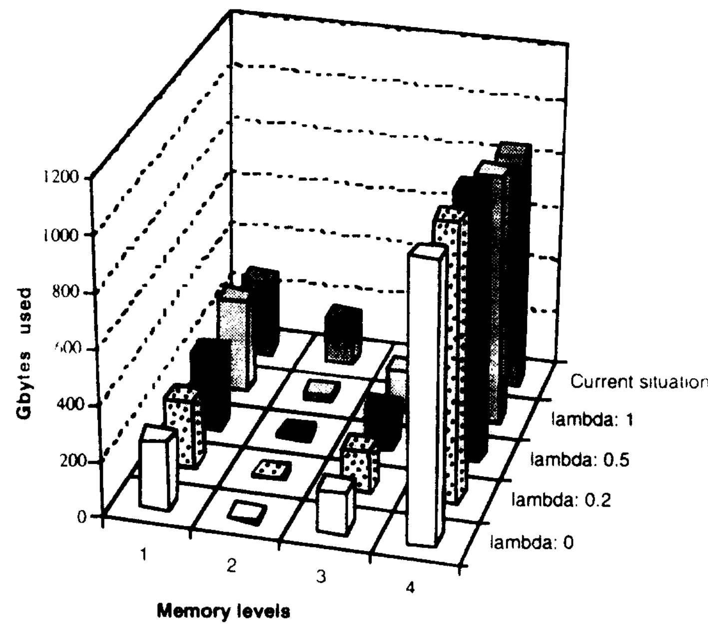

Fig. 5. Uniformly increased by 10%.

4. COMPUTATIONAL RESULTS

4.1. Computational results for the first optimization problem

The first model has been solved for different values of the arrival rates and of the available total memory for the four levels considered. In all the tables in this section, the columns are the memory levels. From left to right: (1) advanced disks, (2) disks, (3) packet data disks, (4) cartridges.

Table 6 gives the optimal values of the proportions for values of $\lambda = \{0.0, 0.2, 0.5, 1.0\}$ and values of b_j corresponding to those of Table 2.

Table 7 shows the optimal distribution for the proportions under hypothesis that the memory capacity is increased uniformly by 10%.

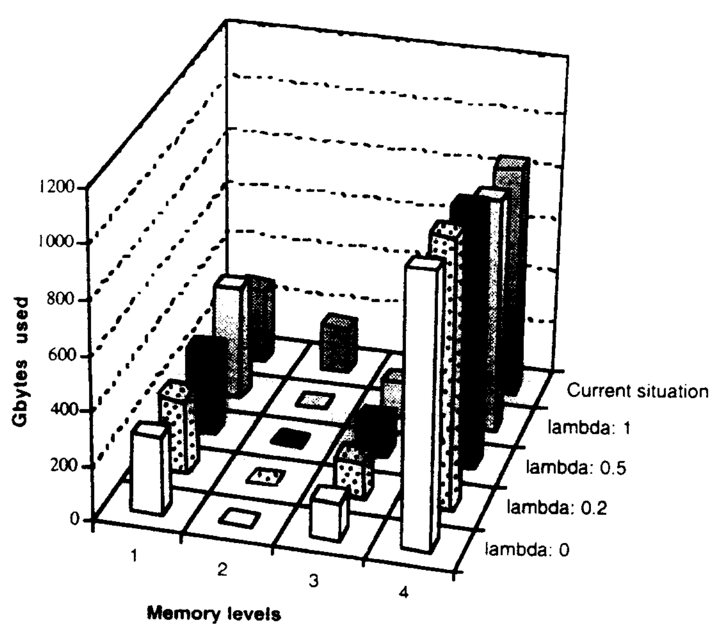

Fig. 6. Level two replaced.

Table 10

	λ = 0				λ = 0.2				λ = 0.5			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.177	0	0.198	0.625	0.369	0	0.032	0.599	0.519	0	0	0.481
Registry data	0.389	0.256	0.354	0	0.542	0.33	0.128	0	0.8	0.2	0	0
Bank book-keeping	0	0	0.2	0.8	0.132	0	0.068	0.8	0.2	0	0	0.8
Transfer orders	0.322	0.368	0.31	0	0.463	0.448	0.089	0	0.721	0.279	0	0
Portfolio	0.117	0	0.664	0.218	0.142	0	0.54	0.318	0.458	0	0.046	0.496
Securities	0.378	0.053	0.569	0	0.533	0.13	0.337	0	0.8	0.2	0	0
Central Bank control data	0.011	0	0.189	0.8	0.111	0	0.089	0.8	0.2	0	0	0.8
External bank data	0.112	0.227	0.661	0	0.191	0.214	0.595	0	0.64	0.134	0.129	0.096
Others	0.184	0	0.016	0.8	0.108	0	0.092	0.8	0.031	0	0.169	0.8

Table 11

	λ = 0.0, 0.2, 0.5				λ = 1			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.551	0	0.449	0	0.8	0.133	0.049	0.018
Registry data	0.235	0.515	0.25	0	0.465	0.535	0	0
Bank book-keeping	0.172	0	0.359	0.469	0.511	0	0	0.489
Transfer orders	0.197	0.577	0.226	0	0.395	0.605	0	0
Portfolio	0.007	0.478	0.515	0	0.277	0.447	0.276	0
Securities	0.151	0.484	0.365	0	0.453	0.547	0	0
Central Bank control data	0.014	0	0.186	0.8	0.2	0	0	0.8
External bank data	0	0.658	0.342	0	0.109	0.778	0.112	0
Others	0.198	0	0.002	0.8	0.044	0	0.156	0.8

Table 8 shows the optimal distribution for the proportion under hypothesis that disks are replaced by advanced disks.

The objective function values for these three different scenarios are given in Table 9.

A graphical representation is drawn in Fig. 3. Here it is possible to observe that a better cost function is obtained under hypothesis of increasing uniformly the memory levels, while worse values are obtained when the second level is completely replaced. This anomaly (we were expecting to decrease the objective function values) is mainly due to a different distribution of the data sets. In fact the portions of data sets memorized on the second levels move on level three and four and to satisfy the time and distribution constraints some of the proportions have to be stored on the first level. This causes an increase in the objective function values. Figures 4, 5 and 6 show the distribution of the data sets in Gbytes for the three different situations described above.

Table 12

	λ = 0.0, 0.2, 0.5				λ = 1			
Checking accounts	0.5	0	0.001	0.499	0.5	0	0.006	0.494
Stocks and bonds	0.403	0.258	0.339	0	0.451	0.269	0.28	0
Registry data	0.149	0.662	0.189	0	0.186	0.691	0.123	0
Bank book-keeping	0.389	0.073	0.538	0	0.427	0.056	0.517	0
Transfer orders	0.128	0.696	0.176	0	0.164	0.726	0.11	0
Portfolio	0	0.766	0.234	0	0	0.739	0.261	0
Securities	0.025	0.729	0.246	0	0.049	0.744	0.207	0
Central Bank control data	0.116	0	0.244	0.641	0.117	0	0.237	0.647
External bank data	0	0.8	0.2	0	0	0.8	0.2	0
Others	0.2	0	0	0.8	0.189	0	0.011	0.8

Table 13

	λ = 0				λ = 0.2				λ = 0.5			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.212	0	0.2	0.59	0.378	0	0.053	0.57	0.523	0	0	0.48
Registry data	0.417	0.24	0.34	0	0.55	0.307	0.143	0	0.8	0.2	0	0
Bank book-keeping	0.006	0	0.19	0.8	0.12	0	0.08	0.8	0.2	0	0	0.8
Transfer orders	0.347	0.35	0.3	0	0.47	0.424	0.106	0	0.682	0.32	0	0
Portfolio	0.147	0	0.67	0.18	0.16	0	0.576	0.26	0.395	0	0.165	0.44
Securities	0.408	0.03	0.56	0	0.539	0.097	0.363	0	0.8	0.2	0	0
Central Bank control data	0.017	0	0.18	0.8	0.101	0	0.099	0.8	0.2	0	0	0.8
External bank data	0.138	0.2	0.66	0	0.202	0.186	0.612	0	0.55	0.16	0.249	0.05
Others	0.164	0	0.04	0.8	0.099	0	0.101	0.8	0.021	0	0.179	0.8

Table 14

	$\lambda = 0.0, 0.2, 0.5$				$\lambda = 1$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.567	0	0.433	0	0.8	0.087	0.113	0
Registry data	0.258	0.504	0.238	0	0.459	0.541	0	0
Bank book-keeping	0.205	0	0.365	0.43	0.54	0	0	0.46
Transfer orders	0.219	0.567	0.214	0	0.392	0.608	0	0
Portfolio	0.029	0.461	0.51	0	0.245	0.444	0.311	0
Securities	0.174	0.471	0.355	0	0.438	0.562	0	0
Central Bank control data	0.02	0	0.18	0.8	0.2	0	0	0.8
External bank data	0	0.66	0.34	0	0.086	0.765	0.149	0
Others	0.178	0	0.022	0.8	0.034	0	0.166	0.8

Table 15

	$\lambda = 0.0, 0.2, 0.5$				$\lambda = 1$			
Checking accounts	0.5	0	0.016	0.484	0.5	0	0.017	0.483
Stocks and bonds	0.446	0.204	0.35	0	0.482	0.225	0.293	0
Registry data	0.175	0.642	0.183	0	0.205	0.669	0.127	0
Bank book-keeping	0.435	0	0.565	0	0.465	0	0.535	0
Transfer orders	0.152	0.679	0.169	0	0.181	0.706	0.113	0
Portfolio	0	0.743	0.257	0	0	0.731	0.269	0
Securities	0.053	0.7	0.247	0	0.07	0.719	0.211	0
Central Bank control data	0.118	0	0.258	0.624	0.131	0	0.24	0.629
External bank data	0	0.8	0.2	0	0	0.8	0.2	0
Others	0.194	0	0.006	0.8	0.178	0	0.022	0.8

4.2. Results for the second optimization problem

We are now considering different scenarios for the second optimization model. Here the goal is to minimize the transactions' completion time. The model is solved for values of the arrival rate equal to: $\{0.0, 0.2, 0.5, 1.0\}$ and for three fixed values for the monthly storage cost (Upper_Bound_Cost). The additional hypotheses to replace the second memory level and to increase uniformly all the memory levels are also tested.

4.2.1. Solutions for the current memory level values. Table 10 shows the optimal values for the proportion values when the Upper_Bound_Cost is 257 million Italian lire. The problem is unfeasible for $\lambda = 1$ because the optimal solution for the first model, with the same parameter values gives an optimal cost equal to 302 million Italian lire.

Table 11 gives the optimal proportion values when the cost is increased to 310 million Italian lire. Here the some distribution is obtained for values of the arrival rate equal to 0.0, 0.2 and 0.5. The reason is that the distribution constraints are in all three cases satisfied as inequalities.

Table 16

	$\lambda = 0$				$\lambda = 0.2$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.085	0	0.336	0.579	0.405	0	0.053	0.542
Registry data	0.37	0	0.63	0	0.8	0	0.2	0
Bank book-keeping	0	0	0.2	0.8	0.136	0	0.064	0.8
Transfer orders	0.361	0	0.639	0	0.8	0	0.2	0
Portfolio	0.007	0	0.8	0.193	0.151	0	0.566	0.283
Securities	0.259	0	0.741	0	0.719	0	0.281	0
Central Bank control data	0.132	0	0.068	0.8	0.123	0	0.077	0.8
External bank data	0.2	0	0.8	0	0.318	0	0.682	0
Others	0.2	0	0	0.8	0.113	0	0.087	0.8

Table 17

	$\lambda = 0.0, 0.2$				$\lambda = 0.5$			
Checking accounts	0.5	0	0	0.5	0.5	0	0	0.5
Stocks and bonds	0.583	0	0.417	0	0.594	0	0.406	0
Registry data	0.405	0	0.595	0	0.544	0	0.456	0
Bank book-keeping	0.388	0	0.298	0.314	0.357	0	0.328	0.314
Transfer orders	0.389	0	0.611	0	0.54	0	0.46	0
Portfolio	0.2	0	0.8	0	0.249	0	0.751	0
Securities	0.311	0	0.689	0	0.492	0	0.508	0
Central Bank control data	0.2	0	0	0.8	0.136	0	0.064	0.8
External bank data	0.2	0	0.8	0	0.236	0	0.764	0
Others	0.2	0	0	0.8	0.2	0	0	0.8

Table 18

	$\lambda = 0.0, 0.2$				$\lambda = 0.5$				$\lambda = 1$			
Checking accounts	0.551	0	0	0.449	0.551	0	0	0.449	0.5	0	0.035	0.466
Stocks and bonds	0.633	0	0.367	0	0.633	0	0.367	0	0.73	0	0.27	0
Registry data	0.42	0	0.58	0	0.42	0	0.58	0	0.726	0	0.274	0
Bank book-keeping	0.649	0	0.351	0	0.649	0	0.351	0	0.689	0	0.311	0
Transfer orders	0.402	0	0.598	0	0.402	0	0.598	0	0.726	0	0.274	0
Portfolio	0.2	0	0.8	0	0.2	0	0.8	0	0.37	0	0.63	0
Securities	0.334	0	0.666	0	0.334	0	0.666	0	0.683	0	0.317	0
Central Bank control data	0.441	0	0	0.559	0.441	0	0	0.559	0.315	0	0.146	0.54
External bank data	0.2	0	0.8	0	0.2	0	0.8	0	0.369	0	0.631	0
Others	0.2	0	0	0.8	0.2	0	0	0.8	0.189	0	0.011	0.8

Table 12 gives the optimal proportion values when the cost is increased to 357 million Italian lire. Also in this case the solutions are equal for the values of the arrival rate equal to 0.0, 0.2 and 0.5.

4.2.2. Solutions for the uniformly incremented memory level spaces. Since the model behaves as in the previous scenario the solutions are reported only for completeness, all the comments apply as in the first case. Table 13 resumes the optimal proportions when the Upper_Bound_Cost is 257 million Italian lire. The problem is unfeasible for $\lambda = 1$.

Table 14 shows the optimal solutions when the Upper_Bound_Cost is equal to 310 Italian million lire.

Table 15 resumes the optimal solutions when the Upper_Bound_Cost is equal to 357 million Italian lire.

4.2.3. Solutions with level two replaced. In this section we present the optimal solutions for the second optimization problem, when the disks are not used and replaced by the advanced disks. Table 16 resumes the optimal solution when the Upper_Bound_Cost is equal to 257 million Italian lire. The problem is unfeasible for $\lambda = 0.5$, $\lambda = 1$, in fact the optimal objective value for this scenario given by the first model is 260 million Italian lire.

Table 17 gives the optimal proportions when the Upper_Bound_Cost is equal to 310 million Italian lire. The problem is unfeasible when $\lambda = 1$ and has the same value for λ equal to 0.0 and 0.2.

Table 18 gives the optimal proportions when the Upper_Bound_Cost is equal to 357 million Italian lire. The optimal solution does not change for values of the arrival rates equal to 0.0, 0.2.

To conclude we report the objective function values; Tables 19, 20 and 21 give the values for all the considered scenarios.

5. CONCLUSIONS

In this paper we have focused on the solution of two optimization problems designed to obtain the optimal distribution of the data sets, used for the daily transactions, over the memory levels. The model was instanced using an objected oriented language by our decision makers, here we have only presented its mathematical structure, which may simplify the complexity of the problem, but gives good insights about the strategies to follow.

Table 19. Current memory levels

λ	Upper cost = 270	Upper cost = 310	Upper cost = 357
0.0	1.388408 e + 5	1.211054 e + 5	1.142047 e + 5
0.2	1.391919 e + 5	1.223261 e + 5	1.142047 e + 5
0.5	1.410073 e + 5	1.223200 e + 5	1.142047 e + 5
1.0	Unfeasible	1.233652 e + 5	1.142284 e + 5

Table 20. Uniformly increased memory levels

λ	Upper cost = 270	Upper cost = 310	Upper cost = 357
0.0	1.480800 e + 5	1305182 e + 5	1.218866 e + 5
0.2	1.483506 e + 5	1.305182 e + 5	1.218866 e + 5
0.5	1.498459 e + 5	1.305182 e + 5	1.218866 e + 5
1.0	Unfeasible	1.314071 e + 5	1.219041 e + 5

Table 21. Replaced second memory level

λ	Upper cost = 270	Upper cost = 310	Upper cost = 357
0.0	1.349640 e + 5	1.197391 e + 5	1.130770 e + 5
0.2	1.483506 e + 5	1.197391 e + 5	1.130770 e + 5
0.5	Unfeasible	1.199107 e + 5	1.130770 e + 5
1.0	Unfeasible	Unfeasible	1.139322 e + 5

Acknowledgement—The authors wish to thank an anonymous referee for improving the readability of this paper.

REFERENCES

1. A. Andronico, L. Cossa, M. Gagliardi and C. Spera. A structured model to manage a large number of transactions. Presented at *Third International Workshop on Project Management and Scheduling*, Como, Italy (1992). To appear in *Eur. J. Opns Res.*
2. A. Andronico, M. Gagliardi and C. Spera. B.L.O.O.M.S: a basic language object oriented for modeling system. Presented at *AIEM3 Conference*, Portland Ore. (1992).
3. M. Gagliardi and C. Spera. A model to manage a large number of transactions: a case study. *Computers ind. Engng* **25**, 287–290 (1993).
4. M. Gagliardi and C. Spera. The syntax of BLOOMS, Part I: Introduction, Technical report, University of Siena (1993).
5. A. M. Geoffrion. The formal aspect of structured modeling. *Opns Res.* **37**, 30–51 (1989).
6. M. Chiarucci and D. Peccianti. Personal Communication (1992).
7. B. A. Murtagh and M. A. Saunders. *Minos 5.0 User's Guide*. Technical Report Sol 83-20, Stanford University (1983).
8. K. G. Murty. *Operation research: deterministic optimization methods*. Prentice–Hall, Englewood Cliffs, N.J. (1994).
9. S. M. Ross. *Introduction to Probabilistic Models*. Academic Press, New York (1988).
10. S. M. Robinson. A quadratically convergent algorithm for the general nonlinear programming problems. *Math. Prog.* **24**, 145–156 (1972).