ELSEVIER

Computer Communications 21 (1998) 1452–1461

computer communications

Optimal grouping of components in a distributed system

Athina P. Markopoulou[a,*], Miltiades E. Anagnostou[b]

[a]Computer Systems Lab., Electrical Engineering Department, Stanford University, Stanford, CA 94305, USA
[b]Computer Science Division, Electrical and Computer Eng. Dept., National Technical University of Athens, 15780 Athens, Greece

Received 4 August 1997; received in revised form 14 April 1998; accepted 14 April 1998

Abstract

Traditionally the performance of a distributed system or a telecommunications network is taken into account only in the last steps of its design and it is seen as a final improvement. Recent attempts to incorporate performance considerations in the mainstream design rely on the development of a functional model consisting of entities that must be optimally distributed over a network of physical nodes. The optimal allocation is environment sensitive and probable different environments must be taken into account. Functional entities that are likely to be grouped together in different environments compose the so-called network entities. In this paper the problem of optimally composing network entities is examined. Different variations of the problem have been studied.

1. Introduction

The objective of this paper is to examine a general problem that arises in the context of a telecommunication system design. Design implies: (1) the definition of a set of software and hardware components that co-operate in such a way as to satisfy a list of initial requirements; (2) their placement in the nodes of a communication network.

The second part of this design process is traditionally driven by experience. Nevertheless, the usage of computers and the increased complexity have encouraged more systematic techniques. The problem has been thoroughly studied within the RACE MONET project that aimed at describing a third-generation mobile telephony system (UMTS [1,11]). This methodology is supposed to be general enough for any distributed system. In order to describe the methodology a few terms must be introduced: the smallest (software or hardware) unit is called the functional entity (FE). The FE can be defined as a grouping of service providing functions in a single location and is a subset of the total set of functions required to provide the service. In other words, it is the atomic unit of distribution. Therefore each node is seen as a collection of FEs and the system design as the allocation of the FEs in each physical entity (PE) of the system. The PE is a set of FEs which is mapped onto a single piece of equipment, actually a network node. In order to determine the FEs contained in each PE one must take into account the environment in which the network will operate. Thus, one may formulate a complex optimization problem. Its solution will then determine the optimal allocation of FEs to PEs. This task is not within the scope of this paper.

Nevertheless, a network is bound to operate under diverse environmental (i.e. offered traffic) conditions. Different conditions will give rise to different instances of the optimization problem and different subsequent optimal allocations. Therefore, the composition of optimal PEs depends on the environment. This is a rather unfortunate situation as far as network produce manufacturers are concerned. They would be obliged either to produce a wide variety of PEs (in order to cover different environments) or to produce each FE as a separate PE. The network entity (NE) concept provides the right size for products to be provided by manufacturers: an NE is a module which groups FEs that tend to be co-located under all or the most frequent environmental conditions. By properly selecting NEs one can then construct PEs. In short, NEs are sets of FEs that appear together in different solutions of the optimal allocation problem. It is the problem of properly selecting the NEs that is dealt with in this paper.

The steps involved in the MONET design methodology (MDM) are:

1. write down a list of requirements to be satisfied by the system;
2. transform the requirements into a graph of interacting FEs;
3. allocate the FEs to PEs appropriately;
4. repeat step 3 for different environments (i.e. geographical distribution of users, traffic generated by users, etc.);

* Corresponding author. Tel.: +30 177 22529; fax: +30 177 22534; e-mail: miltos@telecom.ntua.gr

PII S0140-3664(98)00170-4

5. generate appropriate NEs by taking into account the different results of step 3 obtained for different environments.

Note that a rigorous description of the environmental conditions may help in creating a well-defined mathematical problem. In simple cases a linear programming problem may arise for step 3, whereas more detailed capacity constraints and cost factors may create a difficult combinatorial optimization problem. Alternatively, the different allocations may be the product of a less formal design, and step 3 can be based either on experience or on partial optimizations.

In this paper we deal only with step 5, assuming that the results of the previous steps have somehow (i.e. by using either a strict approach or just a good guess) been produced. Therefore, we concentrate on the problem of optimally creating NEs given the optimum allocations of FEs to PEs for different environmental conditions. We present the problem and the results in a more or less strict mathematical way as we think that the details of specific implementation instances would only add unnecessary complexity to our descriptions. Nevertheless, most of this work has been performed within European Union funded projects dealing with third-generation mobile systems and systematic development of services over a fixed and mobile infrastructure (i.e. RACE MONET and ACTS DOLMEN). We also prefer to present the existing bibliography after having rigorously formulated different versions of the problem, so as to associate each past paper with the right problem.

2. Problems statement

In the rest of the paper we will deal with three "layers" of sets, i.e. the FEs, NEs and PEs. An alternative terminology for these sets could be objects (O), basic services (BS) and services (S) respectively, where the notation O, B and S originates.

Let $O=\{o_1,o_2,\ldots,o_n\}$ be the set of FEs (first layer). NEs (second layer) and PEs (third layer) are both considered to be subsets of O. We think of NEs as being "small" subsets of O, used to compose "larger" subsets of O, the PEs. Our purpose is to describe PEs efficiently. However, it is more efficient to describe them in terms of a few groups of FEs rather than directly in terms of a lot of FEs themselves. FEs which usually appear together in all, or in most, of the PEs should be grouped together to form a separate entity (NE) and should be used all together. That is why we introduce the intermediate layer of NEs.

Given the FEs and the PEs, we will try to choose NEs appropriately in order to optimize some criteria.

Problem P1 is the following.

Given the set of FEs O and a collection of PEs $S=\{S_1,S_2,\ldots,S_K\}$, a set of NEs is defined as a collection B of subsets of $O: B=\{B_1,B_2,\ldots,B_m\}$, such that:

- each FE belongs to exactly one NE and
- $\forall\, S_i$ in S, there are some NEs whose union equals $_{S}i$.

The problem is to find a set of NEs, B, of minimum cardinality.

Problem P2 is a relaxed version of P1.

Given O and $S=\{S_1,S_2,\ldots,S_K\}$,a set of NEs is defined as a collection B of subsets of $O: B=\{B_1,B_2,\ldots,B_m\}$, such that:

- each FE belongs to exactly one NE and
- $\forall\, S_i$ in S, there are some NEs whose union is a superset of S_i, but
- this union is not too big in comparison with S_i, according to some appropriate criterion.

The problem is again to find a set of NEs, B, of minimum cardinality.

Problem P3 is the following.

Given the FEs O and the PEs S, a set of NEs B is defined as a collection of subsets of $O: B=\{B_1,B_2,\ldots,B_m\}$, such that:

- $\forall\, S_i$ in S, there are some NEs (probably trivial) whose union equals S_i.The problem is still to find a set of NEs, B, of minimum cardinality.

Note that in this last problem, each FE may belong to more than one NE, unlike P1 or P2. In other words, B is now a cover, not a partition of O. The problems are compared in Table 1.

In the rest of the paper, we will further elaborate on each of the problems.

3. Partitioning the set of FEs

3.1. Problem P1: formulation and analysis

Given:

- the set of FEs $O=\{o_1,o_2,\ldots,o_n\}$ and
- the set of PEs $S=\{S_1,S_2,\ldots,S_K\}\subseteq 2^O$.*Find*:
- a set of NEs $B=\{B_1,B_2,\ldots,B_m\}\subseteq 2^O$

Table 1
Comparing the problems

	Problem P1	Problem P2	Problem P3
We are looking for a B of minimum cardinality	✓	✓	✓
B is a partition of O	✓	✓	
B is a cover of O			✓
Each S_i in S equals the union of some NEs	✓		✓
Each S_i in S is a subset of the union of some NEs		✓	

such that:

(i) the NEs, B, are a partition of O:

$$\bigcup_{i=1}^{m} B_i = O \text{ and } B_i \cap B_j = \varnothing \ \forall i, j \in 1, 2, \ldots, m$$

(ii) each PE equals the union of some NEs:

$$\forall S_k \in S, \ \exists M' \subseteq 1, 2, 3, \ldots, m : S_k = \bigcup_{j \in M'} B_j$$

(iii) the cardinality of B is minimum: $m = \|B\| = \min$ ***Example***: $O = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $S_1 = \{1, 2, 3, 4, 5\}$, $S_2 = \{3, 4, 5, 7, 8\}$, $S_3 = \{3, 6\}$.

The problem can be represented by either of the diagrams in Fig. 1.

A solution of the problem can be constructed as follows. Those FEs (or objects) which belong to exactly the same PEs (or services), $S_{l_1}, S_{l_2}, \ldots, S_{l_r}$ and which do not belong to any other PE, must be assigned to the same NE. For example, in Fig. 1 $\{1, 2\}$ belongs both to S_1 and to no other PE. $\{4, 5\}$ belongs both to S_1, S_3 and to no other PE. $\{7, 8\}$ belongs to S_2 only, $\{6\}$ alone to S_3 and $\{3\}$ is the only one belonging to all S_1, S_2, S_3. Therefore, the set of NEs B is: $\{1, 2\}, \{3\}, \{4, 5\}, \{6\}, \{7, 8\}$, as drawn in Fig. 1 in dotted lines.

By the way we constructed B, it follows that it is the unique solution of the problem. Indeed, constraint (i) is satisfied, as each FE is assigned to one NE either alone or along with other NEs. Constraint (ii) is also satisfied, i.e. the NE composing a PE contains FEs belonging to the PE, again because of the way these NEs were constructed.

Finally, there is no other set B' which satisfies (i), (ii) and smaller than B. A B' satisfying (i) and (ii) and such that $\|B'\| < \|B\|$, should have at least one NE bigger than the corresponding NE of B. Let us say that B' contains an NE of the type $\{3, 7, \ldots\}$, bigger than $\{3\}$ of B. In this case, S_3 should use $\{3, 7, \ldots\}$ to take FE-3, but it would take the useless FE-7 too, and it would not equal the union of its NEs, which violates constraint (ii). Therefore, B' can only contain subsets of the B solution, like $\{7, 8\}$ or ($\{7\}$ and $\{8\}$) which leads to the same or larger number of NEs than B. Therefore, the B we constructed is the one satisfying (i) and (ii), with minimum cardinality.

A more formal way to construct the above solution is the following algorithm. The second diagram is given and we want to define the NEs.

- At the beginning, the list B of NEs is empty.
- For each FE $o_i = 1, 2, \ldots, n$:
- scan all the PEs and find which of them o_i belongs to. Set a label for o_i indicating those its "PE_{S_i}" $= \{S_{i_1}, S_{i_2}, \ldots, S_{i_r}\}$.
- Search in the list of NEs.

  If there is already an NE with a label equal to "PE_{S_i}", then add o_i to this NE
- else

  add a new NE, with label "PE_{S_i}", at the end of the list, which contains only o_i.
- At the end, the list contains the solution.

The above algorithm obviously requires polynomial time. There are n steps, one for each FE. At each step:

- K PEs are scanned.
- We search the list of NEs (worst case: n steps until the end of the list).
- We assign the FE to the appropriate set of the list.In the worst case, we make in total $n(n + k' + 1)$ steps.

3.2. Problem P2: formulation and analysis

The second constraint of problem P1 (''each PE equals the union of some NEs'') is very strict and it often leads to a large number of NEs, of a few FEs each, because we partition O in completely disjoint sets. In that case, it would be more convenient to work directly with sets of first and third layers without introducing the intermediate layer, which tends to be very close to the first one.

In most applications, we are more interested in defining our PEs/services in terms of a few ''large'' NEs rather than in finding their exact partition into NEs. We might, therefore, allow an NE to have more FEs than those required by the calling PEs. We are going to refer to those extra FEs as ''useless'' for the calling PEs. In this sense, P2 is a relaxed version of P1.

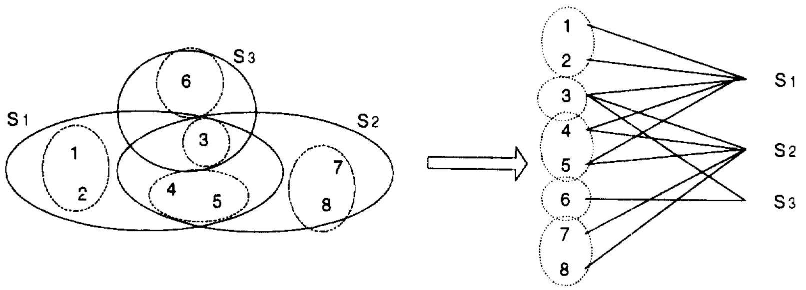

Fig. 1.

On the other hand, we do not want to construct very large NEs by relaxing the constraint. So an upper bound to the number of the useless FEs has to be imposed. It is obvious that this last requirement contradicts both the first one and the requirement for minimum total number of NEs. However, the definition of the intermediate layer would be meaningless without it.

The above ideas, can be expressed formally in problem P2.

Given:

- the set of FEs $O = \{o_1, o_2, \ldots, o_n\}$ an
- the set of PEs $S = \{S_1, S_2, \ldots, S_K\} \subseteq 2^O$.

Find:

- a set of NEs $B = \{B_1, B_2, \ldots, B_m\} \subseteq 2^O$ *such that*:

(i) B should be a partition of O:

$$\bigcup_{i=1}^{m} B_i = O \text{ and } B_i \cap B_j = \varnothing \ \forall i, j \in 1, 2, \ldots, m$$

(ii) If ($S_k' \subseteq 2^O$ is the smallest set: $S_k' \equiv \bigcup_{j\in M'} B_i \supseteq S_k$ where $M' \subseteq \{1, 2, \ldots, m\}$ are indices for the appropriate NEs) then the "useless" (i.e. those belonging to $S_k' - S_k$) must not be much more than the "useful" FEs (i.e. those belonging to S_k), according to some criterion:

$$\|S_k'\| - \|S_k\| \leq \text{err}_k, \ \forall k \text{ or } \frac{\|S_k'\| - \|S_k\|}{\|S_k\|} \leq \text{err}_k\%, \ \forall k$$

where err_k is a given tolerance.(iii) The cardinality of B is minimum: $m = \|B\| = \min$

First, note that if B is a partition of O as (i) demands, then it will be automatically satisfied that "for each S_i in S, there exists a subset of B whose union is a superset of S_i", for example the entire O in the worst case. Then, note that the choice of tolerance err_k, is of great importance for the solution. A small or zero tolerance means that P2 actually turns out to be P1. A very large tolerance means that the whole O tends to be considered as one single NE.

P2 is approached as an optimization problem and, from this point of view, its reduction to the integer programming problem is what naturally comes next.

3.3. Reduction to 0–1 linear programming (LP) problem

This formulation is appropriate for problem P2, which is a typical optimization problem. As far as P1 is concerned, we saw above that it can be solved in polynomial time. Therefore, P1's reduction to the 0–1 LP problem has no other meaning than the uniform formulation of both P1 and P2.

The hypothesis of both problems considers the sets $O = \{o_1, o_2, \ldots, o_n\}$ and $S = \{S_1, S_2, \ldots, S_K\}$ to be given. Therefore, the following numbers can be regarded as given:

$$b_{ik} = \begin{cases} 1, & \text{if } o_i \in S_k \\ 0, & \text{if } o_i \notin S_k \end{cases} \quad i = 1, 2, \ldots, n \quad k = 1, 2, \ldots K$$

We want to choose NEs among all the (non-empty) subsets of $O : \{P_1, P_2, \ldots, P_N\}, N = 2^n - 1$. These subsets are fully described by the following numbers:

$$a_{ij} = \begin{cases} 1, & \text{if } o_i \in P_j \\ 0, & \text{if } o_i \notin P_j \end{cases} \quad i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, N$$

The notation $\{P_1, P_2, \ldots, P_N\}$, for the candidate NEs, should not be confused with the names of our three problems P1, P2, P3.

Among all the candidate subsets $\{P_j\}_{j=1}^{j=N}$, we want to choose our NEs. This choice will be indicated by the variables

$$x_j = \begin{cases} 1, & \text{if } P_j \text{ is chosen to be an NE} \\ 0, & \text{if } P_j \text{ is not chosen} \end{cases} \quad j = 1, 2, \ldots, N$$

The subsets chosen to be our NEs ($x_j = 1$) must be a partition of O, i.e. each FE o_i must belong to exactly one of the NEs:

$$\sum_{j=1}^{N} a_{ij} x_j = 1 \quad \forall i \in \{1, 2, \ldots, n\}$$

In order to write the last constraint of the two problems, let us in addition introduce the following variables.

- It is obvious that $N_{jk} = \Sigma_{i=1}^{n} a_{ij} b_{ik}$ is the number of FEs that candidate subset P_j and PE S_k have in common.
- Then

$$c_{jk} = \begin{cases} 1, & \text{if } N_{jk} > 0 \\ 0, & \text{if } N_{jk} = 0 \end{cases} \quad j = 1, 2, \ldots, N \quad k = 1, 2, \ldots, K$$

indicate whether P_j and S_k have any FEs in common.

- It is also easy to see that $\|P_j\| = \Sigma_{i=1}^{n} a_{ij}$ is the number of FEs belonging to the candidate subset P_j, or "size" of P_j and that $\|S_k\| = \Sigma_{i=1}^{n} b_{ik}$ is the size of PE S_k.

Now, we are ready to write the last constraint for each problem.

According to P1, each PE must *equal* the union of some NEs, i.e. its size must equal the sum of sizes of all the NEs with which it has FEs in common:

$$\sum_{j=1}^{N} x_j c_{jk} \|P_j\| = \|S_k\| \ \forall k \in \{1, 2, \ldots, K\}$$

According to P2, each PE must be a *subset* of the union of some NEs but this union cannot contain many FEs that do not belong to S_k: $\Sigma_{j=1}^{N} x_j c_{jk} \|P_j\| - \|S_k$. So: $\Sigma_{j=1}^{N} x_j c_{jk} \|P_j\| - \|S_k \leq \text{err}_k$, $\forall k = 1, 2, \ldots, K$, where err_k is a given tolerance for each PE S_k.

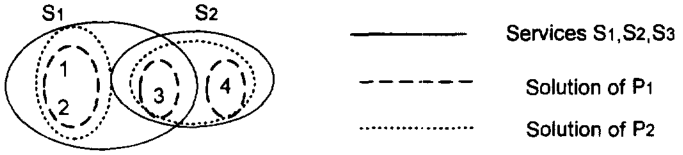

Fig. 2.

Finally, the cost-function to be minimized is the total number of NEs: min $\Sigma_{j=1}^{N} x_j$.

In conclusion, P1 and P2 can be formulated as 0–1 problems as follows.

Problem P1: min $\Sigma_{j=1}^{N} x_j$
subject to:

(i) $\sum_{j=1}^{N} a_{ij}x_j = 1,\ i = 1,\ 2,\ \ldots,\ n$

(ii) $\sum_{j=1}^{N} x_j c_{jk}\|P_j\| = \|S_k\|,\ \forall k \in \{1,\ 2,\ \ldots,\ K\}$

Problem P2: min $\Sigma_{j=1}^{N} x_j$
subject to:

(i) $\sum_{j=1}^{N} a_{ij}x_j = 1,\ i = 1,\ 2,\ \ldots,\ n$

(ii) $\sum_{j=1}^{N} x_j c_{jk}\|P_j\| = \|S_k\| \leq \mathrm{err}_k,\ \forall k = 1,\ 2,\ \ldots,\ K$

The constants $a_{ij}, c_{jk}, \|P_j\|$ and $\|S_k\|$, introduced earlier, are considered to be known, as far as the FEs $O = \{o_1, o_2, \ldots, o_n\}$ and the PEs $S = \{S_1, S_2, \ldots, S_K\}$ are given.

3.4. Examples in P1 and P2

Example 1

$S_1 = \{1,2,3\}, S_2 = \{3,4\}$ (Fig. 2). All (15) subsets of $O = \{1, 2, 3, 4\}$ are candidate NEs.

The solution of P1 in this case is: $\{1, 2\}$, $\{3\}$, $\{4\}$.

P2, with tolerance of 1 useless FE for both PEs, chooses the following NEs: $\{1, 2\}$, $\{3, 4\}$. The NE $\{3, 4\}$ contains FE-4 which is useless for S_1.

Details on the formulation and the code used can be found at the Appendix A.

Example 2

$S_1 = \{1,2,4\}, S_2 = \{3,4\}, S_2 = \{4,5\}$ (Fig. 3).

The solution of P1 in this case is: $\{1,2\}$, $\{3\}$, $\{4\}$, $\{5\}$.

The solution of P2 with tolerance of 1 for S_1, 0 for S_2 and 1 for S_3 is: $\{5\}$, $\{1, 2\}$, $\{3, 4\}$.

The solution of P2 with tolerance of 2 for S_1, 1 for S_2, 1 for S_3 is: $\{1, 2\}$, $\{3, 4, 5\}$.

The solution of P2 with tolerance of 1 for all PEs is: $\{1, 2\}$, $\{4\}$, $\{3, 5\}$.

4. Covering the set of FEs

4.1. Problem P3: formulation and analysis

Unlike P1 and P2, which are looking for a partition of the set of FEs O into NEs, P3 tries to cover the O by NEs. In other words, P3 allows FEs to belong to more than one NE, that is there may be multiple copies of the same FE/object in different nodes. As far as the other constraints are concerned, it is similar to P1. The problem P3 is finally the following.

Given:

- the set of FEs $O = \{o_1, o_2, \ldots, o_n\}$ and
- the set of PEs $S = \{S_1, S_2, \ldots, S_K\} \subseteq 2^O$.*Find*:
- a set of NEs $B = \{B_1, B_2, \ldots, B_m\} \subseteq 2^O$ *such that*:

(i) each PE can be (exactly) composed by some NEs:

$\forall S_k \in S,\ \exists M' \subseteq \{1,\ 2,\ \ldots,\ m\}\ :\ S_k = \bigcup_{j \in M'} B_j$

(ii) the cardinality of B is minimum: $m = \|B\| = \min$.

It should be pointed out that Stockmeyer [2] proved the NP-completeness of the above problem by transformation

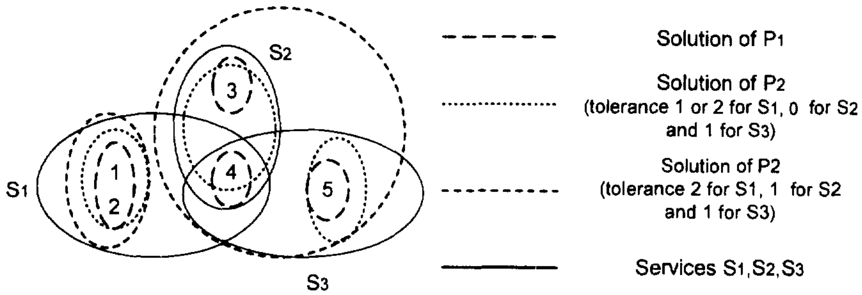

Fig. 3.

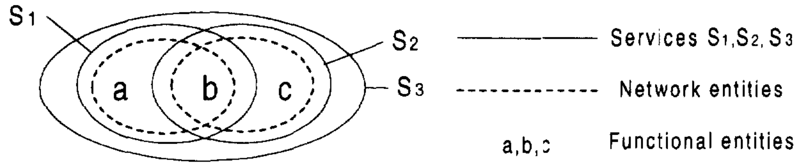

Fig. 4.

from the well-known Vertex Cover Problem [14]. Later, Kou and Wong reduced it to the Clique Cover Problem. Therefore, it makes sense to study heuristics which yield near optimal solutions. In Refs. [3] and [4], the reduction of this problem to another well-known covering problem, the Set Covering Problem, is discussed.

The covering problem [8] is represented by a table, each column of which has to be covered by at least one row. The columns (representing those things to be covered) correspond to elements instances within the PEs $\{S_k\}$. Thus, the table has in total $\|S_1\| + \|S_2\| + \ldots + \|S_{k'}\|$ columns (i.e. the sum of cardinalities of the PEs). For element e_i in S_k there is the corresponding column, which we may call "S_k/e_i". The rows of the table correspond to all possible, non-zero, subsets of OP : $\{P_j\}_{i=1}^{N}$, which are candidates to be chosen for NEs $\{B_j\}$.

The elements of the above table are either 1 or 0 depending on the following criterion. A row representing candidate set P_j is considered to *cover* a column S_k/e_i if and only if: $e_i \in P_j \subseteq S_k$ and the corresponding element of the table will be set 1. Otherwise it will be set 0. That is, P_j covers the element e_i of S_i iff P_j contains $e_i$and is exactly contained by S_k.

4.2. Examples in P3

Let us see some examples.

Example 1

$S_1 = \{a,b\}, S_2 = \{b,c\}, S_3 = \{a,b,c\}$ (Fig. 4).

Table 2 shows the Covering Problem.

Many conventional covering techniques [3,5,6] may now be applied to simplify and finally solve the problem. We can easily see that the solution, in our example, is obviously $B_1 = P_4 = \{a, b\}$ and $B_2 = P_6 = \{b,c\}$.

An obvious disadvantage is the size of the table. However, an important observation, made in Ref. [3], can considerably eliminate the number of rows. The constraint that the sets of the third layer must be exactly composed by sets of the second layer, i.e. $\forall S_k \in S : S_k = \bigcup B_j,\ j \in M' \subset M$, implies that it suffices to regard as candidate sets not all the $2^n - 1$ non-zero subsets of O, but only those which are "exactly contained" in the PEs, i.e. the PEs themselves and their intersections (in combinations of 2, 3, ..., K). In the above example the candidate sets should be the PEs S_1, S_2, S_3 and $\{b\} = S_1 \cap S_2 = S_1 \cap S_2 \cap S_3$ (see Table 3).

Example 2: see Fig. 5.

Instead of all ($2^{10} - 1 = 1023$) possible subsets of $O = \{a,b,c,d,e,f,g,h,i,j\}$it suffices to consider eight candidates sets:

- the PEs:

$$P_1 = S_1 = \{a,b,c,e\}$$
$$P_2 = S_2 = \{e,d,f\}$$
$$P_3 = S_3 = \{g,h,i,j\}$$
$$P_4 = S_4 = \{a,b,c,d,e\}$$
$$P_5 = S_5 = \{f,g,h,i,j\}$$

- the intersections of every two PEs:

$$S_1 \cap S_2 = \{a,b,c,e\} \cap \{e,d,f\} = \{e\} \equiv P_6$$
$$S_1 \cap S_3 = S_1 \cap S_5 = \varnothing$$
$$S_1 \cap S_4 = S_1$$
$$S_2 \cap S_3 = \varnothing$$
$$S_2 \cap S_4 = \{e,d\} \equiv P_7$$
$$S_2 \cap S_5 = \{f\} \equiv P_8$$
$$S_3 \cap S_4 = \varnothing$$
$$S_3 \cap S_5 = S_3$$
$$S_4 \cap S_5 = \varnothing$$

- the intersections of every three, four, five PEs: no new sets are generated.

Table 4 shows the Covering Problem.

Table 2
The Covering Problem table for Example 1

	S_1		S_2		S_3		
	a	b	b	c	a	b	c
$P_1 = \{a\}$	1	0	0	0	1	0	0
$P_2 = \{b\}$	0	1	1	0	0	1	0
$P_3 = \{c\}$	0	0	0	1	0	0	1
$\mathbf{P_4 = \{a,b\}}$	**1**	**1**	**0**	**0**	**1**	**1**	**0**
$P_5 = \{a,c\}$	0	0	0	0	1	0	1
$\mathbf{P_6 = \{b,c\}}$	**0**	**0**	**1**	**1**	**0**	**1**	**1**
$P_7 = \{a,b,c\}$	0	0	0	0	1	1	1

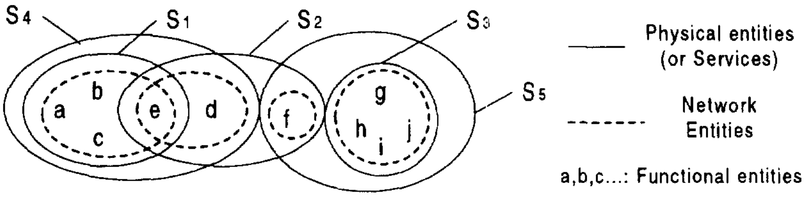

Fig. 5.

In Section 4.3, P3 will be reduced to the 0–1 LP problem and the previous two examples will be solved using 0–1 code.

4.3. Reduction to 0–1 LP problem

P3 has been proved to be in the same class of complexity as the 0–1 LP problem. So, the use of 0–1 code to solve our problem, makes sense in terms of complexity.

It is easy to see that P3 has to be formulated as a 0–1 problem, in a different way than P1 and P2. Indeed, without the partition constraint, the last constraint (that each PE equals the union of some NEs) cannot be written in a linear form. However, P3's formulation becomes straightforward from its reduction to the Covering Problem.

As soon as the table of the Covering Problem is constructed, its elements are known:

$$a_{il} \equiv \begin{cases} 1, & \text{if } e_i \in P_j \subseteq S_k \\ 0, & \text{else} \end{cases} \quad \forall j = 1, \ldots, N', \ l \leftrightarrow k/i$$

Which candidate sets will be finally be chosen to be our NEs is described by the decision-variables:

$$x_j = \begin{cases} 1, & \text{if } P_j \text{ is chosen to be a BS} \\ 0, & \text{if } P_j \text{ is not chosen} \end{cases} \quad j = 1, 2, \ldots, N', \ N' < N$$

We want to cover every column by at least one row:
Problem P3: min $\Sigma_{j=1}^{N'} x_j$
subject to:

$$\sum_{j=1}^{N'} a_{jl} x_j \geq 1, \ \forall \text{ column } l = 1, 2, \ldots \|S_1\|, \ \|S_1\| + 1, \ldots,$$

$$\|S_1\| + \|S_2\|, \ \ldots, \ \|S_1\| + \ldots + \|S_{k'}\|$$

Note that, when we constructed the table of the Covering Problem, we eliminated the candidate sets from $N = 2^n - 1$ to $N' < N$.

We will now apply the 0–1 formulation to the examples of the Section 4.2. The corresponding tables will not be repeated here. The solutions that we find are already drawn in Figs. 4 and 5.

Example 1

In Section 4.2 we studied the problem: $S_1 = \{a, b\}, S_2 = \{b, c\}, S_3 = \{a, b, c\}$.

The reduction is as follows:

min: $x_1 + x_2 + x_3 + x_4$
subject to:

$x_2 > = 1$
$x_1 + x_2 > = 1$
$x_1 + x_3 > = 1$
$x_3 > = 1$
$x_4 + x_2 > = 1$
$x_1 + x_2 + x_3 + x_4 > = 1$
$x_3 + x_4 > = 1$

The solution of the 0–1 problem is $x_2 = x_3 = 1$ and $x_1 = x_4 = 0$ which means that the following sets have been chosen to be our NEs: $P_2 = \{a, b\}, P_3 = \{b, c\}$.

Example 2

The second example of Section 4.2 was

$S_1 = \{a, b, c, e\}$, $S_2 = \{e, d, f\}$,

$S_3 = \{g, h, i, j\}$, $S_4 = \{a, b, c, d, e\}$, $S_5 = \{f, g, h, i, j\}$

The reduction is as follows:

min: $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$
subject to:

$x_1 > = 1$
$x_1 + x_6 > = 1$

Table 3
The simplified Covering Problem table

	S_1		S_2		S_3		
	a	*b*	*b*	*c*	*a*	*b*	*c*
$P_2 = \{b\}$	0	1	1	0	0	1	0
$\boldsymbol{P_4 = \{a, b\} = S_1}$	**1**	**1**	**0**	**0**	**1**	**1**	**0**
$\boldsymbol{P_6 = \{b, c\} = S_2}$	**0**	**0**	**1**	**1**	**0**	**1**	**1**
$P_7 = \{a, b, c\} = S_3$	0	0	0	0	1	1	1

Table 4
The Covering Problem table for Example 2

	S_1				S_2			S_3				S_4					S_5				
	a	*b*	*c*	*e*	*e*	*d*	*f*	*g*	*h*	*i*	*j*	*a*	*b*	*c*	*e*	*d*	*f*	*g*	*h*	*i*	*j*
$P_1 = \{a,b,c,e\}$	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
$P_2 = \{e,d,f\}$	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_3 = \{g,h,i,j\}$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
$P_4 = \{a,b,c,e,d\}$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
$P_5 = \{f,g,h,i,j\}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
$P_6 = \{e\}$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
$P_7 = \{e,d\}$	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
$P_8 = \{f\}$	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

$$x_2 + x_6 + x_7 > = 1$$
$$x_2 + x_7 > = 1$$
$$x_2 + x_8 > = 1$$
$$x_3 > = 1$$
$$x_1 + x_4 > = 1$$
$$x_4 + x_7 > = 1$$

The solution of the 0–1 problem is $x_1 = x_3 = x_7\text{-}yx_8 = 1$ and $x_2 = x_4 = x_5 = x_6 = 0$ which means that the following sets have been chosen to be our NEs: $P_1 = \{a,b,c,e\}, P_3 = \{g,h,i,j\}, P_7 = \{e,d\}, P_8 = \{f\}$.

5. Related work

Our last *problem P3* is known as "The Set Basis Problem". It was clearly stated for the first time in Gimpel's paper on the minimization of spatially multiplexed character sets [3]. From his point of view, Gimpel considered O to be a set of characters and each one of the $\{S_i\}_{i=1}^{i=K}$ to be a character set, such as alphabetics, numerics, alphanumerics. He observed that the character tables, describing the above sets and scanned during the lexical analysis of a source program, needed a large amount of space for storage. So, he proposed that the intermediate layer of sets $\{B_i\}_{i=1}^{i=m}$ should be found and stored instead of the initial character sets $\{S_i\}_{i=1}^{i=K}$ in order to save space.

Similar ideas appear earlier in the work of Bartee [7] and of Pyne and McCluskey [5] in the area of minimization of prime implicant tables. The determination of the simplest sum-of-products expression for a Boolean function can be decomposed into two subproblems. The first one is the determination of the prime implicants and the procedures for their generation. The second subproblem involves the selection of a least-cost subset of these prime implicants, the disjunction of which includes every ONE state of the function (i.e. minterm).

Our problem is somewhat similar to the second subproblem when multiple-output 2-level functions are to be minimized. Each PE/service corresponds to a logical output (second level output or OR gates) and each FE to a minterm. Each NE is a selected set of FEs, in the same way that each prime implicant is a conjunction of minterms (or a set of ONE states of the function). If we consider the minimal expression to be the expression containing the least number of product-terms (i.e. AND gates or prime implicant or first-level outputs), then the similarity to our problem is obvious.

A similar problem also arises in feature extraction and other areas of picture processing. For example, McCluskey [8] noticed that redundancies in a set of patterns can be determined without scanning paths being assumed. From the typical code matrix for the code schedule, he created the pair matrix which corresponds directly to the prime implicant table. Another relative reference, in the same area, is to be found in Ref. [9].

All the above problems could, if realized, be formulated as a covering problem [3], and could finally have been reduced to integer programming [5,8,12,13].

Our first *problem P1* is not clearly mentioned in any of the previous studies. However, it is equivalent to a reduction technique introduced by Gimpel [3], suggested for P3, before this last one is placed in a covering form. In Section 3.1, Section 3.4 we have seen P1 in detail and have proved that it has a unique solution, for which we proposed an efficient algorithm.

Problem P2 differs from P1 only in that it has relaxed the last constraint. So, both P1 and P2 look for a partition of O of minimum cardinality but in P1 each PE must equal the union of some NEs, whereas in P2 it suffices that each PE is a subset of this union. Problem P2 is not mentioned at all in any of the references cited above.

6. Other optimization criteria

In all three problems, we chose the cost-function to be the total number of NEs: $\Sigma_{j=1}^{N} x_j$. However, depending on the application and on the real cost to be minimized, different criteria may be used without other changes in the formulation of our problems.

For example, we could want to minimize the *number of calls to NEs per PE*: (min $\Sigma_{j=1}^{N} x_j c_{jk}$, $k = 1, 2, \ldots, k'$ if and only if NE-j is called by PE-k).

Let us suppose that every time that a PE "calls" an NE, it is billed by the PE machine. Then, the above criterion means that we want to minimize the bill per PE. Its influence to the choice of NEs will be the following.

- Problem P1 has always its unique solution.
- Problem P2 will tend to choose useless FEs, "in the neighborhood of the PE", which is an improvement in its behavior.
- Problem P3 will choose each PE to be an NE, by its own.

Finally, we could choose a cost-function with weights: min $\Sigma_{j=1}^{N} w_j x_j$, where w_j will have a meaning depending on the application. For example, w_j could be some sort of cost for NE-j, or a counter indicating how often this NE-j is called. In all cases, such a criterion will express something like the total cost and it will favor the candidate sets with the least weights-costs.

7. Summary

In the design of a distributed service machine there are several decisions to be made. The service components must be distributed across the nodes of the network so that the desired services are finally implemented/provided by calling the appropriate nodes. We also care about minimizing the network traffic due to such functions.

In this paper we studied three problems which arose during this design process. The first problem creates a partition of the FEs into NEs which are able to exactly compose the PEs (or services). The second problem is a relaxed version of the first one, i.e. the NEs are allowed to contain FEs useless for some PEs, in order to eliminate the total number of NEs. The third problem creates a cover of the FEs by NEs which must exactly compose the PEs.

We clearly stated the above problems and we explored the ideas behind their constraints and optimization criteria. We saw their complexity and we formulated them as 0-1 problems. We also used 0-1 code [10] to solve several examples and we compared the solutions of the problems. We reviewed similar problems in areas completely different from the field of communications, from where our interest initially arose.

Acknowledgements

This work has been partially funded by the ACTS AC036-DOLMEN project. The views of the authors presented do not necessarily reflect the view of the project.

Appendix A Example 1, Appendix 3.4. Details on the formulation and solution using 0-1 code

First of all, we had to calculate the coefficients a_{ij}, c_{jk}, $\|P_j\|$, $\|S_k\|$ (see Table 5). Then we wrote P1, P2 as 0-1 problems:

/* **problem P1**, written as ILP */
/* objective function: min number of NEs */
min: x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + x15;
/* Partition: each FE must belong to exactly one NE */
x1 + x5 + x6 + x7 + x11 + x12 + x13 + x15 = 1;

Table 5
Example 1, Section 3.4

	S_1		S_2		
	N_{k1}	C_{k1}	N_{k2}	c_{k2}	$\|P_j\|$
$P_1 = \{1\}$	1	1	0	0	1
$P_2 = \{2\}$	1	1	0	0	1
$P_3 = \{3\}$	1	1	1	1	1
$P_4 = \{4\}$	0	1	1	1	1
$P_5 = \{1, 2\}$	2	1	0	0	2
$P_6 = \{1, 3\}$	2	1	1	1	2
$P_7 = \{1, 4\}$	1	1	1	1	2
$P_8 = \{2, 3\}$	2	1	1	1	2
$P_9 = \{2, 4\}$	1	1	1	1	2
$P_{10} = \{3, 4\}$	1	1	2	1	2
$P_{11} = \{1, 2, 3\}$	3	1	1	1	3
$P_{12} = \{1, 2, 4\}$	2	1	1	1	3
$P_{13} = \{1, 3, 4\}$	2	1	2	1	3
$P_{14} = \{2, 3, 4\}$	2	1	2	1	3
$P_{15} = \{1, 2, 3, 4\}$	3	1	2	1	4

x2 + x5 + x8 + x9 + x11 + x112 + x14 + x15 = 1;
x3 + x6 + x8 + x10 + x11 + x13 + x14 + x15 = 1;
x4 + x7 + x9 + x10 + x12 + x13 + x14 + x15 = 1;
/* applications S1, S2 must be composed exactly by some NE */
/* The union of NE S1′ must have exactly three FEs (S1 has three FEs) */
x1 + x2 + x3 + 2x5 + 2x6 + 2x7 + 2x8 + 2x9 + 2x10 + 3x11 + 3x12 + 3x13 + 3x14 + 4x15 = 3;
/* The union of NE S2' must have exactly 3 FEs (S2 has 3 FEs) */
x3 + x4 + 2x6 + 2x7 + 2x8 + 2x9 + 2x10 + 3x11 + 3x12 + 3x13 + 3x14 + 4x15 = 2;Solution: $\mathbf{x_3} = \mathbf{x_4} = \mathbf{x_5} = 1$

/* Relaxed **problem P2**, written as ILP */
/* objective function: min number of NEs */
min: x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + x15;
/* Partition: each FE must belong to exactly one NE */
x1 + x5 + x6 + x7 + x11 + x12 + x13 + x15 = 1;
x2 + x5 + x8 + x9 + x11 + x12 + x14 + x15 = 1;
x3 + x6 + x8 + x10 + x11 + x13 + x14 + x15 = 1;
x4 + x7 + x9 + x10 + x12 + x13 + x14 + x15 = 1;
/* The union of NE may contain one FE (= an FE which does not belong to the corresponding PE */
/* |S1′|-|S1| < = 1 */
x1 + x2 + x3 + 2x5 + 2x6 + 2x7 + 2x8 + 2x9 + 2x10 + 3x11 + 3x12 + 3x13 + 3x14 + 4x15 < = 4;
/* |S2′|-|S2| < = 1 */
x3 + x4 + 2x6 + 2x7 + 2x8 + 2x9 + 2x10 + 3x11 + 3x12 + 3x13 + 3x14 + 4x15 < = 3;Solution: $\mathbf{x_5} = \mathbf{x_{10}} = \mathbf{1}$

References

[1] Towards the Personal Communication Environment: Green Paper on Mobile and Personal Communications, European Commission DGXIII, April 27, 1994.

[2] L.J. Stockmeyer, The set basis problem is NP-complete, IBM Report RC-5431, May, 1975.

[3] J.F. Gimbel, The minimization of spatially multiplexed character sets, Commun. ACM 17 (6) (1974) 315–318.

[4] L.T. Kou, C. K. Wong, A note on the set basis problem related to the compaction of character sets, IBM Report RC-5244, January, 1975.

[5] I.B. Pyne, E.J. McCluskey, An essay on prime implicants tables, J. SIAM 9 (1961) 604–631.

[6] J.F. Gimbel, A reduction technique for prime implicant tables, IEEE Trans. 14 (4) (1965) 535–541.

[7] T.C. Bartee, Computer design of multiple-output logical networks, IRE Trans. 10 (2) (1961) 21–30.

[8] E.J. McCluskey, Determination of redundancies in a set of patterns, IRE Trans. Inf. Theory 3 (1957) 167.

[9] H.D. Block, N.J. Nilsson, R.O. Duda, Determination and detection of features in patterns, in: J.T. Tou, R.H. Wilcox (Eds.), Computer and Information Sciences, Spartan Books, Washington, DC, 1964, pp. 75-110.

[10] P. Barth, A Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization, Max-Planck-Instut fuer Informatik, January, 1995 (+ OPBDP package: ftp.mpi-sb.mpg.de:pub/guide/staff/barth/opbdp/).

[11] CEC deliverable R2066/BT/PM2/DS/P/070/b2, UMTS System Structure Document.

[12] R.S Garfinkel, G.L. Nemhauser, Integer Programming, Wiley-Interscience, New York, 1972.

[13] F. Afrati, G. Papageorgiou, Algorithms: Design Methods and Complexity Analysis, National Tech. Univ. Athens Press, Athens 1993 (in Greek).

[14] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, New York, 1979.