분산데이타 할당모형의 라그랑지안 문제 생성시스템

이재규(KAIST)
김철수(원광대)
정민화(원광대)

분산데이타 할당 모형의 라그랑지안 문제 생성시스템

Generator of Lagrangian Problem for Distirbuted Data Allocation Models

이 재 규 (한국과학기술원 테크노경영대학원)
김 철 수, 정민화 (원광대학교 정보관리학과)

1. INTRODUCTION

In a field of model management, one of the major research points is the representation for user's domain problems. Since representing the general problem is not only difficult but also impractical, a group of researchers has narrowed its focus on the representation of a specific category such as optimization problems (Geoffrion, 1989; Krishnan, 1989; Lee and Kim, 1995; Liang, 1988). So far, the research scope was the semantic representation of optimization model and the aiding model formulation process under the assumption that one of the standard solvers such as simplex algorithm and branch and bound algorithm can be used.

For overcoming the problems, we develop a UNIK-RELAX system being a generator of Lagrangian problem by structural identification and relaxation approach (Kim and Lee, 1996). The approach includes three issues such as representation of embedded structures, identification of embedded structures, and relaxation algorithm. This paper describes the system UNIK-RELAX that implements the proposed approach.

For this purpose, we shall first the review semantic representation for IP model and Lagrangian problem in UNIK-OPT (Lee and Kim, 1995; Lee et al., 1989b) with a generalized assignment problem in Section 2. To transform the model into Lagrangian problems, it is necessary to represent the IP model as three states: model distinctiveness state, embedded structure state, and Lagrangian structure state. So a knowledge-assisted modeler UNIK-OPT is adopted for this purpose in Section 3. Section 4 shows the approach for finding distinctiveness of a embedded structure in the IP model where the approach includes bottom up approach, and we suggest the relaxation algorithm of IP model to an adequate Lagrangian problem. In Section 5, we show an overall architecture of the UNIK-RELAX developed and the important components composed of the system. In Section 6, the system UNIK-RELAX is applied to one of the data allocation model.

2. SEMANTIC REPRESENTATION FOR IP MODEL AND LAGRANGIAN IN UNIK-OPT

This section explains semantic representation for IP model and Lagrangian problem in UNIK-OPT. We describe several views for explaining the approach. As a good example for illustrating the representation, we use the generalized assignment problem, which is rich with readily apparent structure. First, we will show the semantic represntation for Lagrangain problem being transfored from IP models.

The generalized assignment problem (GAP) is the integer programming named "plant assignment problem";

The generalized assignment problem (GAP) is the integer programming named "plant assignment problem";

$$Minimize \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad \text{......................(6)}$$

subject to

$$\sum_{i=1}^{m} a_{ij}x_{ij} \le b_j, \quad j = 1, \ldots, n \qquad \text{......................(7)}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m \qquad \text{......................(8)}$$

$$x_{ij} = \{0, 1\} \quad \forall_{ij} \qquad \text{......................(9)}$$

We can represent this problem using the semantic view of UNIK. The semantic view specifies the optimization model using formal knowledge representation such as frames (Lee and Kim, 1995). Figure 1 is a frame representation of plant assignment problem.

The UNIK-OPT supports three levels of views: semantic, notational (aggregate or individual equational), and tabular views. The model shown in (6)-(9) corresponds to an aggregate notational view. The semantic view should at least include more information than notational view so that it can be transformed to the notational view without any ambiguity. In a UNIK-OPT system, the semantic IP model is represneted by objects being automatically transformed to the mathematical notational form.

Dualizing second constraints (Fisher, 1981), we can get Lagrangian problem with real number as follows:

$$Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} \lambda_i(\sum_{j=1}^{n} x_{ij} - 1)$$

subject to

$$\sum_{i=1}^{m} a_{ij}x_{ij} \le b_j \quad j = 1, \ldots, n$$

$$x_{ij} = \{0, 1\} \quad \forall_{ij}$$

As such, we can relax the IP problem into the Lagrangian problem by dualizing particular constraints. But, in order to perform this process, we need OR-expert. Moreover, even though good formulation on a modelling process is accomplished, it takes much effort to link the formulated model to general solver. Branch and Bound algorithm. Since only good formulation does not guarantee easy acquisition of the solution, it is necessary for model management system to encompass representations of solution procedure like Lagrangian relaxation. By doing so, it can deal with various complex domains and support many user.

3. THREE STATES FOR GENERATING LAGRANGIAN PROBLEMS

We perform automatic structural identification and relaxation procedure based on frame representation like Figure 1. In this paper, we will specify three states by contents of frame. Three states are equal to representation types in a semantic view.

- Model Distinctiveness State
- Embedded Structure State

```
{{ plant_assignment_problem
        IS-A : IP_MODEL
        DIRECTION : min
        OBJECTIVE : (+ total_cost_BOT)
        CONSTRAINT : plant_capacity_constraint
                     plant_assignment_constraint}}
{{ plant_capacity_constraint
        IS-A : CONSTRAINT
        OPERATOR : LE
        LHS : (+ plant_sum_BOT)
        RHS : (+ plant_capacity_BOT)
        UNIT_ INDEX : plant }}
{{ plant_assignment_constraint
        IS-A : CONSTRAINT
        OPERATOR : EQ
        LHS : (+ plant_choice_BOT)
        RHS : (+ one_BOT)
        UNIT_INDEX : product }}
{{ total_cost_BOT
        IS-A : BOT
        ATTRIBUTE : unit_cost assignment_var
        SUMMATION_INDEX : product plant }}
{{ plant_sum_BOT
        IS-A : BOT
        ATTRIBUTE : plant_volume assignment_var
        SUMMATION_INDEX : product }}
{{ plant_capacity_BOT
        IS-A : BOT
        ATTRIBUTE : plant_capacity
        SUMMATION_INDEX : }}
{{ plant_choice_BOT
        IS-A : BOT
        ATTRIBUTE : one assignment_var
        SUMMATION_INDEX : plant }}
{{ one_BOT
        IS-A : BOT
        ATTRIBUTE : one
        SUMMATION_ INDEX : }}
{{ assignment_var
        IS-A : VARIABLE
        SYMBOL : x
        LINKED_INDEX : product plant
        TYPE : binary }}
{{ unit_cost                              {{ plant_volume
        IS-A : COEFFICIENT                        IS-A : COEFFICIENT
        SYMBOL : c                                SYMBOL : a
        LINKED_INDEX : product plant }}           LINKED_INDEX : product plant }}
{{ plant_capacity                         {{ one
        IS-A :COEFFICIENT                         IS-A :COEFFICIENT
        SYMBOL : b                                SYMBOL : 1
        LINKED_INDEX : plant}}                    LINKED_INDEX : }}
{{ product                                {{ plant
        IS-A : INDEX                              IS-A : INDEX
        SYMBOL : i                                SYMBOL : j
        LINKED_ATTRIBUTE : assignment_var         LINKED_ATTRIBUTE : assignment_var
          unit_cost plant_volume }}                 unit_cost plant_volume plant_capacity }}
```

FIGURE 1. Frame Representation of Plant Assignment Problem in the UNIK-OPT Syntax

- Lagrangian Problem State

The relationships among them are graphically shown in Figure 2.

3.1. Model Distinctiveness State

The model distinctiveness state means that all frames have distinctiveness (We will explain in Section 3) which is used for describing the characteristic of embedded structures. The objects in UNIK-OPT consist of *variables* and *coefficients*, *blocks of terms* (BOT : a pair of variable and coefficient that shares the same summation sign), *constraint chunks* (each of which consists of BOTs and an operator of EQ, GE, or LE), and a *specific model* (BOTs in the objective function and a set of constraint chunks). To endow a frame per se with distinctiveness, we should start from database level.

So, to endow attribute object with distinctiveness, we get coefficients from the database and variables of attribute from user. To endow BOT objects with distinctiveness, we must use attribute objects obtained before. As such, we take bottom up approach which identifies the distinctiveness for each objects in the order of attribute, BOT, constraint, and model. For example, in plant assignment problem, distinctiveness views of object per se are represented as follows :

- Attribute

{{ assignment_var
IS-A : VARIABLE
SYMBOL : x
LINKED_INDEX : product plant
DISTINCTIVENESS : binary
}}
{{ unit_cost
IS-A : COEFFICIENT
SYMBOL : c
LINKED_INDEX : product plant
DISTINCTIVENESS : nonnegative
}}

- BOT

{{ total_cost_BOT
IS-A : BOT
ATTRIBUTE : unit_cost assignment_var
SUMMATION_INDEX : product plant
DISTINCTIVENESS : volume_sum_0_1
}}

- Constraint

{{ plant_capacity_constraint
IS-A : CONSTRAINT

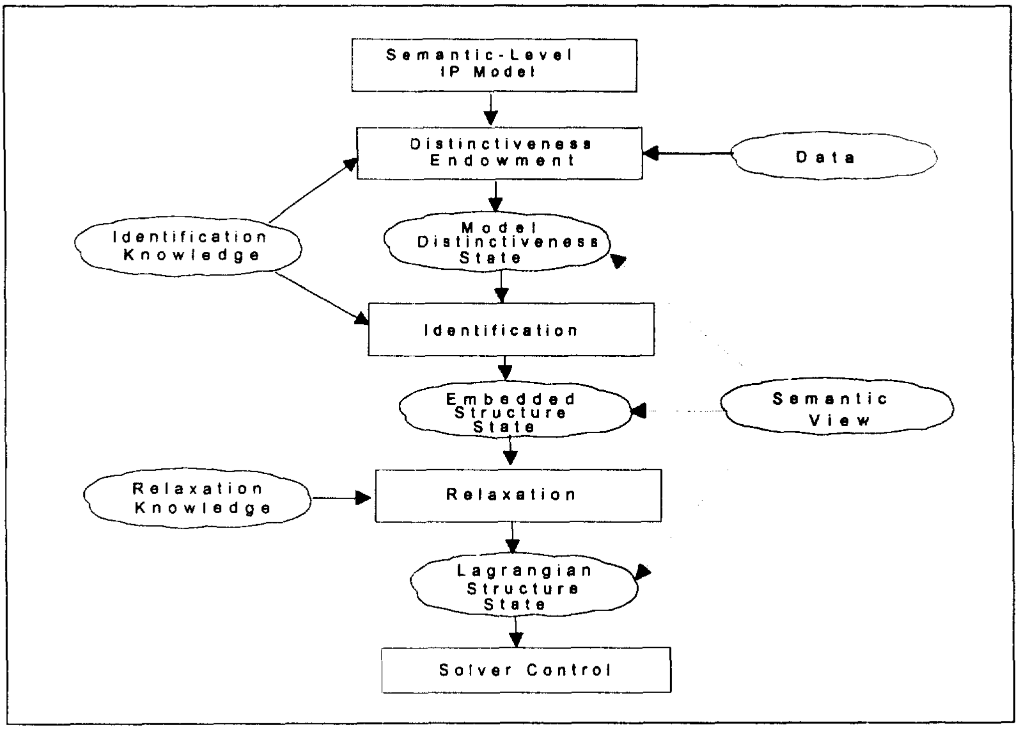


FIGURE 2. Three Structural Identification and Relaxation View

```
(RULE volume_sum_0_1_BOT
 IF
  (BOT
   ^frame-name <BOT>
   ^ATTRIBUTE (equal (get-value (nth 0 <>) 'DISTINCTIVENESS)
                nonnegative_real_coefficient)
   ^ATTRIBUTE (equal (get-value (nth 1 <>) 'DISTINCTIVENESS) '0_1_integer_variable))
 THEN
  (new-value <BOT> 'DISTINCTIVENESS 'volume_sum_0_1 ))

(RULE nonnegative_real_coefficient_BOT
 IF
  (BOT
   ^frame-name <BOT>
   ^ATTRIBUTE (equal (get-value 'DISTINCTIVENESS) 'nonnegative)
 THEN
  (new-value <BOT> 'DISTINCTIVENESS 'nonnegative_real_coefficient ))

(RULE capacity_constraint
 IF
  (CONSTRAINT
   ^frame-name <CONSTRAINT>
   ^LHS (equal (get-value <> 'DISTINCTIVENESS) 'volume_sum_0_1 )
   ^RHS (equal (get-value <> 'DISTINCTIVENESS) 'nonnegative_real_coefficient )
   ^OPERATOR = 'LE)
 THEN
  (new-value <CONSTRAINT> 'DISTINCTIVENESS 'capacity ))

(RULE 0_1_knapsack_structure
 [structure 0_1_knapsack]
 IF
  (MODEL
   ^frame-name <MODEL>
   ^CONSTRAINT (equal (get-value <> 'DISTINCTIVENESS) 'capacity ))
 THEN
  (new-value <MODEL> 'EMBEDDED_STRUCTURE '0_1_knapsack ))
```

FIGURE 3. Illustrative Rules for the Identification of 0_1_Knapsack Structure in the UNIK-FWD Syntax

OPERATOR : LE
LHS : (+ plant_sum_BOT)
RHS : (+ plant_capacity_BOT)
UNIT_ INDEX : plant
DISTINCTIVENESS : capacity

}}

- Model

{{ plant_assignment_problem
IS-A : IP_MODEL
DIRECTION : min
OBJECTIVE : (+ total_cost_BOT)
CONSTRAINT : plant_capacity_constraint
plant_assignment_constraint
MODEL_STRUCTURE : generalized_assignment

}}

3.2. Embedded Structure State

This state represents one of characteristics of each substructure in the IP model such as *0_1_knapsack* or *0_1_GUB*. There exist efficiency priorities between embedded structures in the IP model (Fisher, 1981). According to the priorities (Which will be mentioned in Section 4), the embedded structure is determined to 0_1_knapsack. To build up the state, we should perform processes of identification using rulebase. Using the identified state, we can generate Lagrangian problem. For example, plant assignment problem, embedded structural state of model object is represented as follows:

- Model

{{ plant_assignment_problem
IS-A : IP_MODEL
DIRECTION : min
OBJECTIVE : (+ total_cost_BOT)
CONSTRAINT : plant_capacity_constraint
plant_assignment_constraint
MODEL_STRUCTURE : generalized_assignment
EMBEDDED_STRUCTURE : 0_1_knapsack

}}

3.3. Lagrangian Structure State

Lagrangian structure state is an output which results from the relaxation procedure. If there is at least one of the seven embedded structures, the original problem can be relaxed into Lagrangian problems (we will describe in Section 5). After relaxation procedure, the problem is connected to solvable heuristics by solver controller. In the previous example, Lagrangian structure state is represented as follows :

- Model

{{ *plant_assignment_problem_RELAXED*
IS-A : *LAGRANGIAN_MODEL*
DIRECTION : min
OBJECTIVE : (+ total_cost_BOT)
(+lambda_plant_choice_BOT)
(-lambda_one_BOT)
CONSTRAINT : plant_capacity_constraint
MODEL_STRUCTURE : 0_1_knapsack
}}

4. MODEL IDENTIFICATION AND RELAXATION ALGORITHM

Note that the objects in Figure 1 do not have the attribute *distinctiveness*. However, to apply the Lagrangian Relaxation, the distinctiveness of all objects should be identified until the one named *EMBEDDED_STRUCTURE* in the IP-MODEL object is identified as shown in Section 3. UNIK-RELAX performs the identification procedure for each object. For this, we employ the bottom-up approach.

4.1. Bottom Up Approach for Model Identification

The bottom-up approach identifies the distinctivenesses in the order of variables and coefficients, BOTs, constraints, and the embedded structure of specific model. For instance, the problem have five attributes. Except for variable attribute, all of them are identified by scanning datatbase. Followings are attributes having distinctiveness identified through scanning database.

{{ unit_cost
IS-A : COEFFICIENT
SYMBOL : c
LINKED_INDEX : product plant
DISTINCTIVENESS : nonnegative
}}
{{ plant_volume
IS-A : COEFFICIENT
SYMBOL : a
LINKED_INDEX : product plant
DISTINCTIVENESS : nonnegative
}}
{{ plant_capacity
IS-A : COEFFICIENT
SYMBOL : b
LINKED_INDEX : product
DISTINCTIVENESS : nonnegative
}}
{{ one
IS-A : COEFFICIENT
SYMBOL : 1
LINKED_INDEX :
DISTINCTIVENESS : constant
}}
{{ assignment_var
IS-A : VARIABLE
SYMBOL : x
LINKED_INDEX : product plant
DISTINCTIVENESS : nonnegative
}}

After accomplishing identification, with distinctivenesses of attributes, the identification procedure of BOTs is performed. Distinctivenesses of BOTs' are as followings;

```
{{ plant_sum_BOT
                IS-A  :  BOT
                ATTRIBUTE : plant_volume_assignment_var
                SUMMATION_INDEX : product
                DISTINCTIVENESS : nonnegative_real_coefficient
}}
{{ total_cost_BOT
                IS-A  :  BOT
                ATTRIBUTE : unit_cost_assignment_var
                SUMMATION_INDEX : product plant
                DISTINCTIVENESS : volume_sum_0_1
}}
{{ plant_capacity_BOT
                IS-A  :  BOT
                ATTRIBUTE : plant_capacity
                SUMMATION_INDEX :
                DISTINCTIVENESS : nonnegative_real_coefficient
}}
{{ plant_choice_BOT
                IS-A  :  BOT
                ATTRIBUTE : one_assignment_var
                SUMMATION_INDEX : plant
                DISTINCTIVENESS : count_sum_0_1  first_XOR_second_index_sum
}}
{{ one_BOT
                IS-A  :  BOT
                ATTRIBUTE : one
                SUMMATION_INDEX :
                DISTINCTIVENESS : constant_1_coefficient
}}
```

After accomplishing identification of BOTs with distinctivenesses of BOTs, the same way is applied to identification of constraint chunks. Distinctivenesses for constraint chunks are as followings;

```
{{ plant_capacity_constraint
                IS-A  :  CONSTRAINT
                OPERATOR : LE
                LHS : (+ plant_sum_BOT)
                RHS : (+ plant_capacity_BOT)
                UNIT_INDEX :  plant
                DISTINCTIVENESS : capacity
}}
{{ plant_assignment_constraint
                IS-A  :  CONSTRAINT
                OPERATOR : EQ
                LHS : (+ plant_choice_BOT)
                RHS : (+ one_BOT)
                UNIT_INDEX :  product
                DISTINCTIVENESS : 0_1_GUB
}}
```

To implement this approach, a forward chaining tool like UNIK-FWD (Lee et al., 1994) can be used. In Figure 3, illustrative rules for the identification of *0_1_Knapsack* structure are represented in the UNIK-FWD syntax. The first two rules identify the *volume_sum_0_1* and *nonnegative_real_coefficient* BOTs, the third rule for the *capacity* constraint, and the last rule for the *0_1_Knapsack* structure.

4.2. RELAXATION ALGORITHM OF IP MODELS INTO LAGRANGIAN PROBLEMS

As mentioned in Section 2, we deal with relaxation of IP models into Lagrangian problems on semantic view. First, we explain concept of the semantic view transformation : attribute and BOT generation, objective function update, and object elimination, and we proceed five phases relevant to the relaxation procedure using the concept. The concept will be used in the representation of solution procedures for adopting knowledge-assisted schemes to OR methodology such as partitioning method.

The relaxation algorithm is performed as frame transformations on a semantic view. We need to describe the characteristic of the semantic view and frame transformations relevant to relaxation of IP models into Lagrangian problems.

- **Attribute and BOT Generation**

We have to generate new frame, attribute and BOT, after identifying embedded structure. For instances, the new frame are lambda attribute and BOTs having lambda attribute in Attribute slot value. Process of generation is very simple as followings;

- **Objective Function Update**

In case that generating and updating of the frames in IP model reflect the *OBJECTIVE* slot of *IP_MODEL*. For such transformations, it is necessary to update *the OBJECTIVE slot value* with other frames such as *lambda* and *BOT_by_lambda*.

- **Object Elimination**

In case of relaxation procedure, there is some elimination of constraint objects to be relaxed and BOT objects its exclusively associated BOTs. For instances, the constraint *constraint_relaxed* and its exclusively associated BOTs BOT_exclusive.

After having identified embedded structure in the model (or structure of the model per se), UNIK-RELAX relaxes the problem to a Lagrangian problem as described in Section 2. The relaxation can be performed by the following five steps. The relaxed Lagrangian problem of the generalized assignment problem in Figure 1 is a *0_1_Knapsack* problem. Note the added BOTs and modified constants by the multipliers are in shades, while the eliminated ones are in the solid rectangles.

Let us examine these steps with the plant assignment problem (Kim and Lee, 1995).

Step 1. Identify the constraint objects to be relaxed.

Step 2. Generate new attribute for Lagrangian multipliers in the lambda objects.

Step 3. Generate new BOTs that incorporate the Lagrangian multipliers in the coefficients.

Step 4. Add the Lagrangian BOTs to the OBJECTIVE of the IP-model object.

Step 5. Eliminate the identified constraint(s) to be relaxed in step 1 and its exclusively associated BOTs.

5. DEVELOPMENT OF UNIK-RELAX

UNIK-RELAX implements the suggested procedure of automatic structural identification and relaxation as a part of the UNIK (UNIfied Knowledge) system. UNIK-RELAX is under development using the UNIK environment. The relevant capabilities employed from UNIK are objects (UNIK-OBJECT), forward chaining (UNIK-FWD) and backward chaining (UNIK-BWD) reasoning, and integer programming (UNIK-OPT) model representation and formulation aid (Lee et al., 1994).

UNIK-RELAX represents an IP model semantically using UNIK-OBJECT so that the IP model can communicate with other knowledge bases. Thus, this IP model formulation can be treated as an extraction of relevant knowledge from the knowledge base.

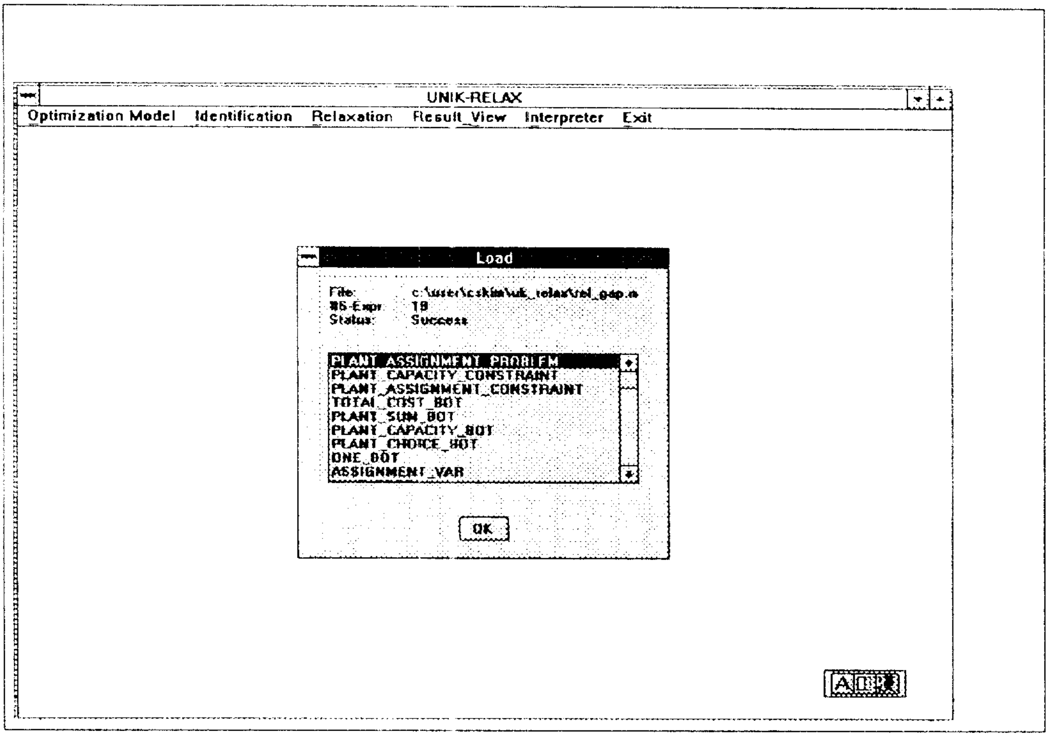
This section illustrates the system of UNIK-RELAX. The systems have four imporant subsystems : **"IP selector"**, **"Embedded Structures Identifier"**, **"Lagrangian Model Transformer"**, and **"Solver Controller"**. We show the solution procedure of automatic structural identification and relaxation with illustrative screens.

Before the solution procedure, some IP models are represented as the semantic view by UNIK-OPT syntax and the rule base in UNIK-FWD and UNIK-BWD syntax is prepared. As previously explained in Section 2, we will show the procedure as examples of generalized assignment problem named as *plant_assignment_problem*.

- IP Model Selector

This section shows illustrative screens for main menu, model selection, model frame description, and model mathematical form. After having an open menu of the integer model, if the user inputs or selects an IP model file, then the semantic view of the model is shown by frame as Figure 4.

Figure shows this name of frames composed of the selected model such as plant_capacity_constraint, total_cost_BOT, and assignment_var. If user have knowledge about content of BOTs frame, user may select the menu - ***Identification -> Identified Model -> BOT Object***. Among the frames, plant_assignement_problem (*IS-A : IP MODEL*) frame are viewed. This semantic view can be transformed to mathematical notational form. This model have five attribute objects, five BOT objects, two constraint chunk objects, and model object.



File: c:\user\cskim\uk_relax\rel_gap.m
#S-Expr 18
Status: Success

PLANT_ASSIGNMENT_PROBLEM
PLANT_CAPACITY_CONSTRAINT
PLANT_ASSIGNMENT_CONSTRAINT
TOTAL_COST_BOT
PLANT_SUM_BOT
PLANT_CAPACITY_BOT
PLANT_CHOICE_BOT
ONE_BOT
ASSIGNMENT_VAR

FIGURE 4. Frame Name Screen for Selected IP Model

- Embedded Structure Identifier

Embedded structure identifier have three steps as follows;

- Select of identification rule

- Load the rule
- Apply the rule to selected model for identifying embedded structures

This section show illustrative screens for three steps. After opening a menu of the rule, if user select a rule, contents considered in the rule are loaded. And if user clicks the menu - ***Identification* -> *Run to Identify*,** UNIK-RELAX applies the rule to the selected IP model. After identifying embedded structures, user can know the results of identification which are shown by clicking the menu - ***Identification* -> *Identified Model* -> (*Model Object, Constraint Object, BOT Object, Attribute Object*).**

- Lagrangian Model Transformer

Lagrangian Model Transformer have two steps as follows;

- Select of Relaxation rule, and load the rule
- Apply the rule to the identified model for relaxation to Lagrangian problem

This section show illustrative screens for two steps. The system can show generated attribute (*lambda*) and two BOTs (lambda_plant_choice_BOT and lambda_one_BOT), and the relaxed Lagrangian problem.

- Solver Controller

The *solver controller* generates the adequate numbers for the Lagrangian multipliers and associates the primal solution algorithm (Branch and Bound) with the computed bounds by the Lagrangian problems.

6. APPLICATION TO DATA ALLOCATION MODEL

The data allocation problem is one of determining how to allocate parts of a global database to the computing sites connected by means of a network. The data allocation design is an essential factor affecting the efficiency and effectiveness of a distributed database in meeting geographically dispersed database processing demands. There are three generic data allocation models applicable to a variety of system environments: one model for performance-based data allocation designs in LANs or some MANs under weak locality of reference (Model LM-W-P), another model for cost-based designs in MANs or WANs under weak locality of reference (Model MW-W-C), and a third model for cost-based data allocations in some MANs and WANs under strong locality of reference (Model MW-S-C) (Sheng and Lee, 1992).

Among three models, third model includes various substructures in structural point of view. So we concentrated on a special model named of WAN-value-added which inherits the structure of the model MW-S-C.

6.1 WAN-value-added

An optimization model, WAN-value-added for data allocation design in a value-added WAN inherits the structure of the model MW-S-C. Only communication costs and storage constraints are included in this model. Therefore, the optimization model now has the following formulation.

$$\mathit{Minimize} \sum_{i=1}^{m} \sum_{j=1}^{n} [\sum_{k \neq i} c_{lki} y_{lki} + \sum_{k \neq i} d_{lki} x_{li}]$$

subject to

$$\sum_{i} x_i = 1,$$

$$y_{lki} \leq x_{li}, \quad \forall_{l,k,i,k \neq i}$$

$$\sum_{l=1}^{n} a_l x_{lj} \leq b_i, \quad \forall_i$$

$$\sum_{k \neq i}^{m} y_{lki} + x_{lj} = 1, \quad \forall_{l,j}$$

$$x_{lj}, \quad y_{lkj} = \{0, 1\} \quad \forall_{li}$$

where

c_{lkj} : The total query communications cost to access database fragment l from node k to node j.

d_{lki} : The total update communications cost to access database fragment from node k to node j,

a_l : The storage requirement of database fragment l,

b_i : The storage capacity limit at node i.

$$x_{lj} = \begin{cases} 1, \text{ if file l is allocated to node j} \\ 0, \text{ otherwise} \end{cases}$$

$$y_{lkj} = \begin{cases} 1, \text{ if query traffic from node k for file l is routed to node j} \\ 0, \text{ otherwise} \end{cases}$$

The model described here shares characteristics common to many existing models (Casey, 1972; Dowdy and Foster, 1982; Ramamoorthy, 1983). Decision variables x_{lj} that represent the allocation of each fragment to network nodes are defined.

A feasible solution needs to satisfy routing feasibility in addition to capacity constraints. We assume the files should not be replicated in any node. Specifically, a query request will be served locally if a copy of the referenced fragment resides at the requesting site; otherwise, the request will be routed to some node where the fragment is located. Thus,

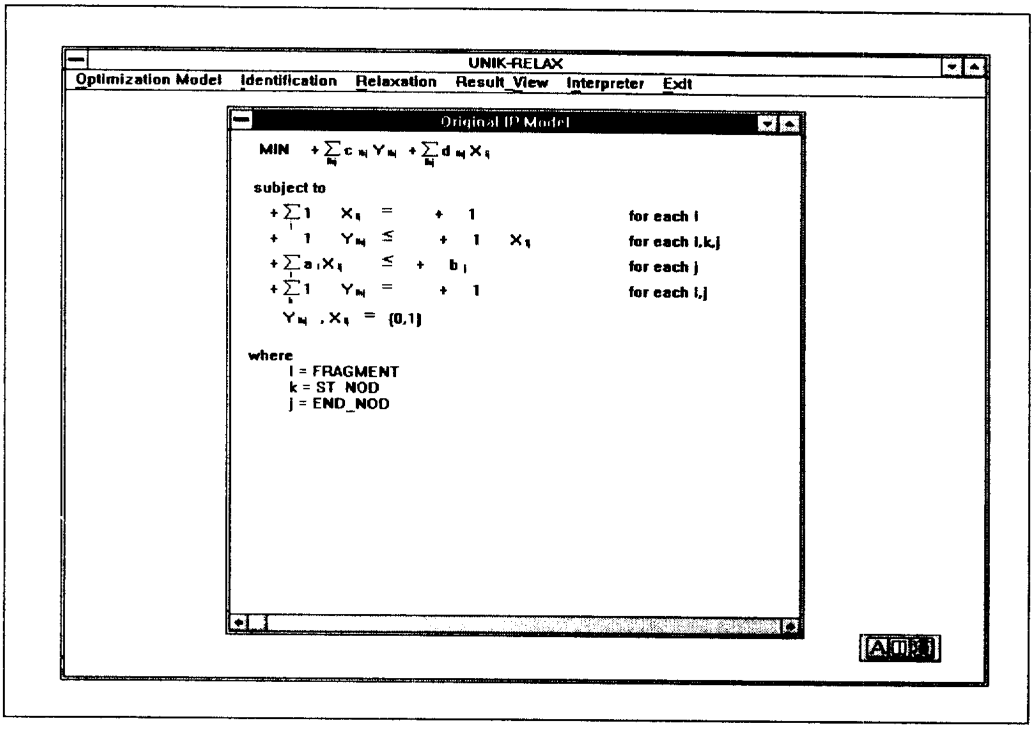
$$\sum_{i \neq k} y_{lki} + x_{lk} = 1, \quad \forall_{l=1,\ldots,F,k=1,\ldots,N}$$

Since a request from node k can be routed to node i only if a copy of the file exist there, it is required that

$$y_{lki} \leq x_{ik}, \quad \forall_{l=1,\ldots,F,k=1,\ldots,N,i=1,\ldots,N,k \neq i}.$$

It has been shown that model MW-S-C or a similar formulation requires NP-complete computational complexity. However, in most cases, the linear programming relaxation of this model gives integer solutions.

The UNIK-RELAX system is applied to the data allocation model *(WAN-value-added)*. First, we represent the data allocation model as semantic view, and we apply automatic structural identification and relaxation to Lagrangian problem for the model.



MIN $+\sum_{lkj} c_{lkj} Y_{lkj} + \sum_{lj} d_{lj} X_{lj}$

subject to

$+\sum_{j} 1 \quad X_{lj} = + 1$ for each l

$+ 1 \quad Y_{lkj} \leq + 1 \quad X_{lj}$ for each l,k,j

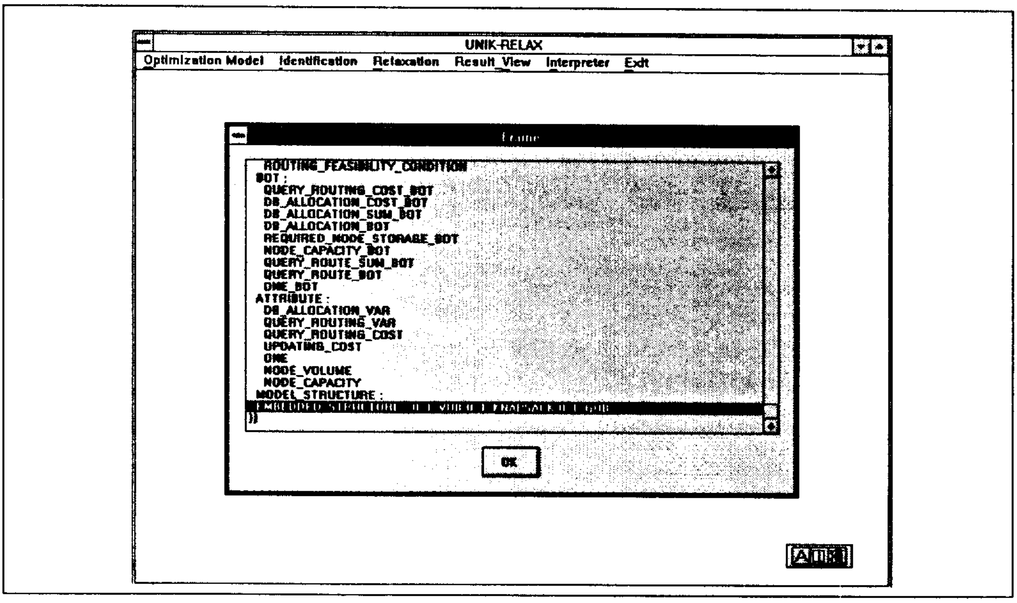
$+\sum_{l} a_{l} X_{lj} \leq + b_{j}$ for each j

$+\sum_{k} 1 \quad Y_{lkj} = + 1$ for each l,j

$Y_{lkj}, X_{lj} = \{0,1\}$

where
l = FRAGMENT
k = ST_NOD
j = END_NOD

FIGURE 5. Mathematical Notational Representation



```
  ROUTING_FEASIBILITY_CONDITION
 BOT :
  QUERY_ROUTING_COST_BOT
  DB_ALLOCATION_COST_BOT
  DB_ALLOCATION_SUM_BOT
  DB_ALLOCATION_BOT
  REQUIRED_NODE_STORAGE_BOT
  NODE_CAPACITY_BOT
  QUERY_ROUTE_SUM_BOT
  QUERY_ROUTE_BOT
  ONE_BOT
 ATTRIBUTE :
  DB_ALLOCATION_VAR
  QUERY_ROUTING_VAR
  QUERY_ROUTING_COST
  UPDATING_COST
  ONE
  NODE_VOLUME
  NODE_CAPACITY
 MODEL_STRUCTURE :
 EMBEDDED_STRUCTURE [illegible] VUB [illegible] KNAPSACK [illegible] GUB
}}
```

FIGURE 6. Embedded Structures of the Data Allocation Model

6.2. Illustrative Screens of UNIK-RELAX

We represented the model named of *data allocation problem* as semantic view in section 7.1. We show illustrative screens from applying UNIK-RELAX to the model. The model, *data allocation problem*, has two constraints chunk objects, nine BOT objects , and seven attribute objects. This model can be transformed to mathematical notational form as Figure 5. And embedded structure identification proceeds as following steps : selecting identification rule, loading the rule and applying the rule to selected model. After applying the model, UNIK-RELAX endows each objects with distinctiveness as in Figure 6.

After applying the embedded structures, UNIK-RELAX relaxes the model to Lagrangian model with the structure. The Lagrangian problem is illustrated in Figure 7. And mathematical notational form is shown as in Figure 8.

7. CONCLUSION

We have seen that UNIK-RELAX can support the transformation of IP models into Lagrangian problems. Since UNIK-RELAX can identify the distinctiveness of embedded structure for the IP model having complex strcuture, the model should be relaxed into the Lagrangian problem. UNIK-RELAX may be also applied to real domain such as data allocation model design (Liu Sheng, 1992) and power generation systems (Muckstadt and Koenig, 1977). Currently, the prototype of UNIK-RELAX runs on the MS-WINDOWS. The research in the Intelligent Information Systems Laboratory at KAIST seeks not only to represent the solution procedure as structural identification of IP models and their relaxation into the Lagrangian model but also to unify the rules and optimization models (Lee and Song, 1995), rules and constraint satisfaction problems, and optimization models and neural networks to establish a framework of namely Unified Programming (Lee, 1995).

REFERENCES

Casey, R. G., Allocation of copies of a file in an information network, *Proc. AFIPS Spring Joint Computer Conf.*, Vol. 40 (AFIPS Press, Montvale, NJ, 1972)

Dowdy, L. W. and D. V. Foster, Comparative models of the file assignment problem, *Comput. Survey* 14 (1982) 287-313.

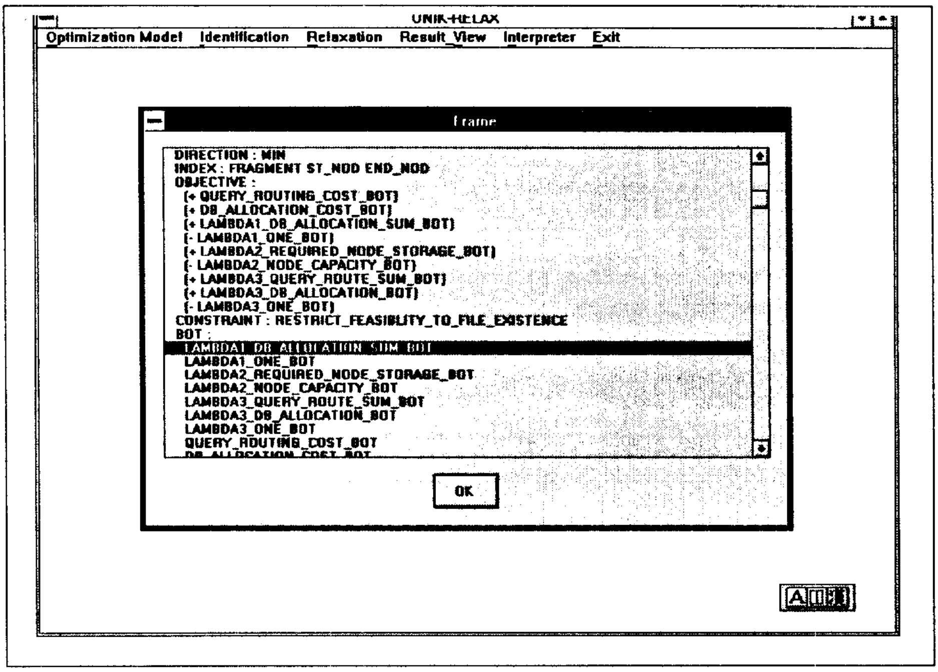
Fisher, M. L., The Lagrangian relaxation method for solving integer programming problems. *Management Science*, Vol. 27, No. 1, (Jan 1981).

Geoffrion, A. M., The Formal Aspects of Structured Modeling, *Operations Research* 37, No.1, (1989).

Kim, Chulsoo and Jae K. Lee, Automatic Structural Indetification and Relaxtion for Integer Programming, *Decision Support Systems* 18(3&4), (1996)

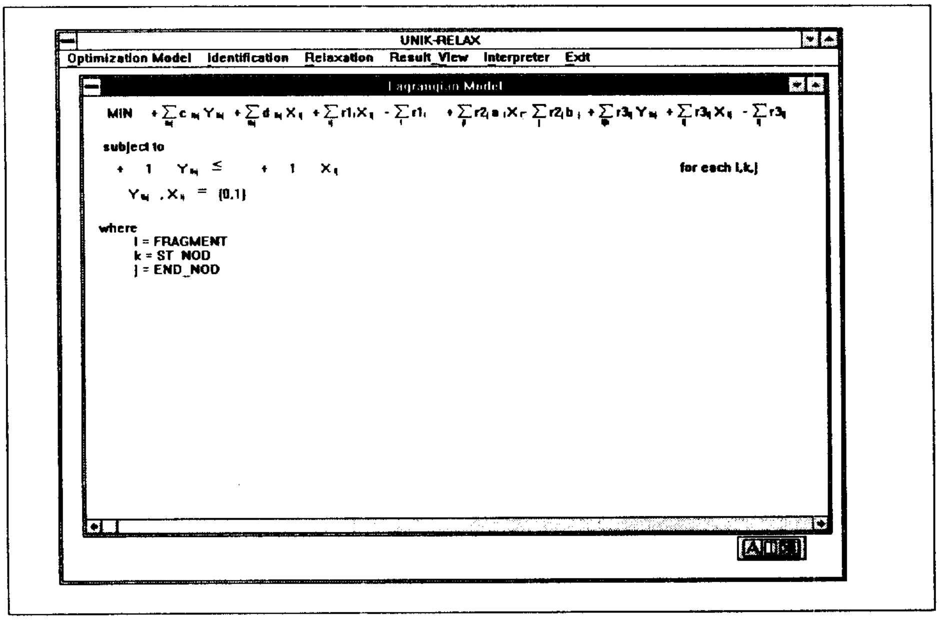
Krishnan, R., X. Li, and D. Steier, A Knowledge-Based Mathematical Model Formulation System, *Communications of the ACM* 35, No.9, (1992).

Lee, Jae K. and M.Y. Kim, Knowledge-Assisted Optimization Model Formulation: UNIK-*OPT*, *Decision Support Sytems* 13, (1995).



```
DIRECTION : MIN
INDEX : FRAGMENT ST_NOD END_NOD
OBJECTIVE :
 [+ QUERY_ROUTING_COST_BOT]
 [+ DB_ALLOCATION_COST_BOT]
 [+ LAMBDA1_DB_ALLOCATION_SUM_BOT]
 [- LAMBDA1_ONE_BOT]
 [+ LAMBDA2_REQUIRED_NODE_STORAGE_BOT]
 [- LAMBDA2_NODE_CAPACITY_BOT]
 [+ LAMBDA3_QUERY_ROUTE_SUM_BOT]
 [+ LAMBDA3_DB_ALLOCATION_BOT]
 [- LAMBDA3_ONE_BOT]
CONSTRAINT : RESTRICT_FEASIBLITY_TO_FILE_EXISTENCE
BOT :
  LAMBDA1_DB_ALLOCATION_SUM_BOT
  LAMBDA1_ONE_BOT
  LAMBDA2_REQUIRED_NODE_STORAGE_BOT
  LAMBDA2_NODE_CAPACITY_BOT
  LAMBDA3_QUERY_ROUTE_SUM_BOT
  LAMBDA3_DB_ALLOCATION_BOT
  LAMBDA3_ONE_BOT
  QUERY_ROUTING_COST_BOT
```

FIGURE 7. Lagrangian Model of the Data Allocation Model



$$\mathrm{MIN}\ \ +\sum_{lkj} c_{lkj} Y_{lkj} + \sum_{lj} d_{lj} X_{lj} + \sum_{lj} r1_l X_{lj} - \sum_{l} r1_l \ \ + \sum_{lj} r2_j a_l X_{lj} - \sum_{j} r2_j b_j + \sum_{lkj} r3_{lj} Y_{lkj} + \sum_{lj} r3_{lj} X_{lj} - \sum_{lj} r3_{lj}$$

subject to

$+\ 1\ \ Y_{lkj} \le\ \ +\ 1\ \ X_{lj}$ for each l,k,j

$Y_{lkj}\ ,X_{lj} = \{0,1\}$

where
l = FRAGMENT
k = ST_NOD
j = END_NOD

FIGURE 8. Mathematical Notational Representation of Lagrangian Model

Lee, Jae K. and Y. Song, Unification of Linear Programming with a Rule-based System by the Post-Model Analysis Approach, *Management Science* 41, No. 9, (1995).

Lee, Jae K. et al. UNIK User Manual, Intelligent Information System Laboratory in *Korea Advancd Institute of Science and Technology*, (1994).

Liang, T. P., Development of a Knowledge-based Model Management System, *Operations Research* 36, No.6 (1988).

Liu Sheng, O.R. and H. Lee, Data Allocation Design in Computer Networks : LAN versus MAN versus WAN, *Annals of Operation Research*, 36, (1992) 125-150.

Muckstadt, J. A. and S. A. Koenig, An Application of Lagrangian Relaxation to Scheduling in Power Generation Systems, *Operations Res.* Vol 25 (May-June 1977)

Murphy, F. H., E.A. Stohr, and P. Ma, Composite Rules For Building Linear Programming Models From Component Models, *Management Sciences* 41(9), (1995)

Ramamoorthy and B. W. Wah, The isomorphism of simple file allocation, *IEEE Trans. Comput.* C-32 (1983).