정수계획 모형에서 라그란지안 구조정의 및 완화를 지원하는 지능형 시스템의 개발
(Development of an Intelligent System for Lagrangian Structural Identification and Relaxation for Integer Programmings)

김 철 수*, 이 재 규**, 김 민 용***

* 경민전문대학 사무자동화과 조교수

** 한국과학기술원 경영정보공학과 교수

*** 경희대학교 경영학과 조교수

Abstract

This research investigates the automatic identification of typical embedded structures in the Integer Programming (IP) models and automatic transformation of the problem to an adequate Lagrangian problem which can provide tight bounds within the acceptable run time. For this purpose, the structural distinctiveness of variables, constants, blocks of terms, and constraint chunks is identified to describe the structure of the IP model. To assist the identification of the structural distictiveness, the representation by the knowledge based IP model formulator UNIK-IP is adopted. For the reasoning for the structural identification, the bottom-up, top-down, and case-based approaches are proposed. A prototype system UNIK-RELAX is developed to implement the approaches proposed in this research.

1. 개요

수리계획 모형은 다양한 응용분야에서 경영의사결정을 지원하기 위해서 사용되었다. 그러나, 모형생성의 지식집약적 (knowledge- intensive) 특성과 모형유지의 필요성이 더욱 더 비전문적인 사용자들에게는 모형 사용을 어렵게 만들었다. 이를 극복하기 위해서 의사결정지원시스템 (Decision Support Systems) 연구자들은 모형관리 (Model Management) 분야에 큰 관심을 쏟고 연구해 왔다. 그들이 연구해 온 초점 중에 하나는 대상 문제의 표현이었다 [17, 22, 25, 29, 30].

일반적인 대상 문제는 표현하기 어려울 뿐만아니라 적용하기는 더욱 어렵다. 그래서 표현하고자 하는 대상을 최적화 문제와 같은 특정한 부문에 맞추었다. 모형관리 분야는 처음에 선형계획 모형을 표현하는 연구이었으며, 이를 정수계획

모형과 비선형 계획 모형에까지 확장하고 있다. 이러한 연구의 초점은 지식기반을 이용해 대상 문제를 의미론적인 표현에 중점을 두었으며, 해법에서도 Simplex 알고리즘, Interior Point 알고리즘 및 Branch and Bound 알고리즘 하에서의 해를 찾는 대상문제에만 연구가 이루어졌다 [25, 34].

이제는 정수계획 모형에서 NP-hard 특성을 갖는 문제를 표현하고, 그 문제에 라그란지안 완화기법 (Lagrangian Relaxation Method) 을 적용하여 모형 운영에 전문가가 아닌 의사결정자를 돕고[20], 그 대상문제를 해결함으로써 풀기 어려운 대상문제에 대해 쉽게 접근하도록 만드는 것이 모형관리 분야의 시급한 과제이다.

이러한 과제를 해결하기 위해서, 우선 정수계획 문제에서 NP-hard의 문제 유형과 그 유형에 적용하여 훌륭한 결과를 갖는 라그란지안의 모형을 정의하는 과정이 필요하며, 그 선행과정으로 정수계획 모형의 내포구조 (Embedded Structure) 를 발견하는 과정이 필요하다. 그 내포구조는 크게 7 가지로 분류된다. 0-1 knapsack 구조, assignment 구조, generalized upper bound 구조, variable upper bound 구조, one spanning tree 구조, 동적계획 모형의 속성을 갖는 knapsack 구조, 일반적인 정수계획 구조 들이다. 사용자는 정수계획 모형생성 시스템인 UNIK-IP 를 이용해서 모형을 의미론적인 관점에서 표현을 하게되며, 그 표현된 모형의 내포 구조를 발견하게 되는데, 그 구조는 네가지의 객체로 구성된다. 그 네가지의 객체는 변수(Variables) 및 계수 (Coefficients) 객체, 항 (Blocks Of Terms: BOT's) 객체, 제약식 (Constraint Chunks) 객체, 그리고 목적식 (Objective Function) 객체들이다. 각 객체 별로 Distinctiveness 를 갖는다. 다음에는 각 구조가 사용자에 의해서 표현된 정수계획 모형에서 정의하는 추론과정이 필요하다. 이러한 추론과정은 크게 3가지로 분류된다. 모형의 변수 및 계수 객체로부터 추론해 나가는 정방향 추론 방식 (Forward Chaining Method), 내포 구조의 특성부터 정의하여 하향식 (top-down) 으로 내려가며 객체 특성을 정의하는 역방향 추론 방식 (Backward Chaining Method), 그리고 과거에 추론되어진 경험을 갖고 내포구조의 객체를 정의하는 경험적 추론 방식 (Case Based Method) 들을 이용한다. 내포구조가 정의되면 라그란지안 모형으로 완화시키는 변환하는 과정이 필요한데 이는 UNIK-IP 상에서 의미론적 표현에서 이루어진다. 이 과정은 크게 5가지로 나눌 수가 있는데 먼저 제약식 객체를 발견하고, 그 제약식에 관계된 라그란지안 승수를 새로운 객체로 생성되며, 그 승수에 의해서 새로운 항 (BOT) 을 생성한다. 그 후에 목적식에 승수와 변형된 제약식이 결합하여 더해지는 후에 완화된 제약식 객체와 라그란지안 모형에 이용되지 않는 항 객체들과 계수 객체들은 삭제된다.

위의 이론적인 개념을 바탕으로 프로토타입 시스템인 UNIK-RELAX (UNIfied Knowledge -RELAXation) 가 개발되었다. 그 시스템은 객체 지식표현시스템 (UNIK-OBJECT), 정 방향추론 시스템 (UNIK-FWD), 역 방향 추론시스템 (UNIK-BWD), 정수계획 모형화 시스템 (UNIK-IP) 등과의 Interface를 갖는다 [24].

2. 라그란지안 완화 (Lagrangian Relaxation) 기법

다음과 같은 정수계획 모형 (1) - (3) 은 풀기 어려운 제약식 $g_i(\mathbf{x}) \geq 0$ 을 포함하고 있다고 가정하자.

$$\textit{Minimize} \quad z(\mathbf{x}) \quad \text{.....(1)}$$
$$\textit{subject to} \quad g_i(\mathbf{x}) \geq 0 \quad i = 1, ..., r \text{..(2)}$$
$$\textit{and} \quad \mathbf{x} \in X \quad \text{.....(3)}$$

새로운 라그란지안 모형은 제약식 (3)을 가지고 목적함수는

$$L(\mathbf{x}, \mathbf{u}) = z(\mathbf{x}) + \Sigma_{i=1}^{r} u_i g_i(\mathbf{x})$$

을 갖는다. 그리고 제약식 (2)에 해당하는 라그란지안 승수를 u_is 하자. 그러면 라그란지안 완화기법에 의해서 생성된 모형은 다음과 같다.

$$\textit{Minimize} \quad L(\mathbf{x}, \mathbf{u}) = z(\mathbf{x}) + \Sigma_{i=1}^{r} u_i g_i(\mathbf{x})$$
$$\textit{subject to} \quad \mathbf{x} \in X$$

Fisher [9] 는 NP-hard 문제들을 Generalized Assignment Problem, General Integer Programming Problem, Set Covering and Partitioning Problem, Location Problem, Scheduling Problem, and Traveling Salesman Problem 등의 6가지 유형으로 분류하고 이러한 문제에 라그란지안 완화기법을 적용하여 좋은 결과를 얻은 라그란지안 문제를 설명하고 있다.

즉, 정수계획 문제가 위의 표에서 나타난 7가지의 유형 중에 하나를 포함하고 있으면 옆의 열에 해당하는 라그란지안 문제로 완화가 되는 것을 설명하고 있다. 그리고 완화되는 제약식은 내포구조에 따라 다르다. 그 규칙을 설명하면 다음과 같다.

No	Embedded Structure	Lagrangian Problem
1	*0_1_Knapsack*	0_1_Knapsack problem
2	*Assignment (Perfect-2 Matching)*	Assignment problem (Perfect-2 Matching)
3	*0-1 General Upper Bound(GUB)*	0-1 General Upper Bound problem
4	*0-1 Variable Upper Bound(VUB)*	0-1 Variable Upper Bound problem
5	*1_Spanning_tree*	1-Spanning tree problem
6	*Knapsack_with _time _horizon*	Pseudo polynomial Dynamic Programming (DP)
7	*General_IP_with _unbounded*	Group problem

<Table 1> Relationships between Embedded Structure and Lagrangian Model

1) 내포구조가 1-5 인 경우 : 그 구조와 결합되지 않은 제약식을 완화한다.

2) 내포구조가 6 인 경우 : 그 구조와 결합된 제약식을 완화한다.

3) 내포구조가 7 인 경우 : 본래의 모형에서 unbounded variables 를 갖는 제약식 모두를 완화한다.

3. 내포구조 (Embedded Structures) 의 표현

제3장에서는 전형적인 내포구조를 정의하기 위해 필요한 특징적인 요소를 표현하고자 한다. UNIK-IP [34] 는 정수계획 모형을 4가지의 객체를 가지고 표현하고 있다. 이들 객체는 내포구조의 특성을 묘사하기 위해 여러 속성 (attribute) 을 갖는데, 그것을 우리는 Distinctiveness 라고 부른다.

그러면 4가지의 객체를 설명하고 전형적인 7가지의 내포구조가 각 객체로 어떻게 표현되는 가를 설명해보자.

3.1 변수와 계수 (Variables and Coefficients)

변수에 대한 Distinctiveness 는 *binary, nonnegative_integer,* 그리고 *integer* 등 으로 구분하고, 계수의 Distinctiveness 는 *real, nonnegative, nonpositive, constant.* 로 구분한다.

3.2 항 (Blocks of Terms:**BOT**) 의 Distinctiveness

BOT 의 Distinctiveness 는 기본적인 것과 복합적인 것으로 나눌 수가 있다.

1) 기본적인 Distinctiveness

- *Integer_Variable:* each variable in the BOT is an integer variable.
- *0_1_Integer_Variable:* each variable in the BOT is either *0* or *1*.
- *Nonnegative_Integer_Variable:* each variable in the BOT is a nonnegative integer variable.
- *Real_Coefficient*: each coefficient in the BOT is a real number.
- *Nonnegative_Real_Coefficient*: each coefficient in the BOT is a nonnegative real number.
- *Constant_1_Coefficient*: each coefficient in the BOT is *1*.
- *Constant_2_Coefficient*: each coefficient in the BOT is *2*
- *Constant_n_Coefficient*: each coefficient in the BOT is the same number *n*
- *Constant_S-1_Coefficient:* a value of coefficient in the BOT is the "total number of elements included in the set S" minus 1.
- *Two_Indices:* there exist two summation indices in the BOT.
- *Precedence_Restricted_Index*: the first index in the BOT has a precedence restriction over the second one in the *Two_Indices* cases.
- *Sum_with_First_Index:* the BOT is summed by the first index.
- *Sum_with_Second_Index:* the BOT is summed by the second index.

- *Time_Indexed:* one of the indices in the BOT implies time unit.
- *First_Index_is_1:* all values of the first index in the BOT are fixed to 1.

2) 복합적인 Distinctiveness

기본적인 Distinctivenesses 는 AND, OR, 또는 XOR 와 같은 연산자를 이용하여 다양한 복합적인 Distinctivenesses 를 구성할 수가 있다. 일반적인 형태는 다음과 같다.

OPERATOR (*OPERATOR*

(*Elementary_distinctiveness_1*

Elementary_distinctiveness_2)

Elementary_distinctiveness_3)

- *Volume_Sum_0_1* = *AND* (*Nonnegative_Real_Coefficient* *0_1_Integer_Variable*)
- *Volume_Sum_Integer* = *AND* (*Nonnegative_Real_Coefficient* *Nonnegative_Integer_Variable*)
- *Count_Sum_0_1* = *AND* (*Constant_1_Coefficient* *0_1_Integer_Variable*)
- *Precedence_Node* = *AND* (*Precedence_Restricted_Index* *0_1_Integer_Variable*)
- *Starting_Node* = *AND* (*AND* (*First_Index_is_1* *Sum_with_Second_Index*) *0_1_Integer_Variable*)
- *First_XOR_Second_Index_Sum* =*AND* (*XOR* (*Sum_with_First_Index* *Sum_with_Second_Index*) *Two_Indices*)

3) BOT 의 Distinctiveness 들의 포함관계

이러한 포함관계는 하나의 BOT 가 여러 Distinctiveness 를 갖는 경우에, 혹은 여러개의 BOT 들을 갖는 제약식 객체에서 정의하기 위해 선행작업으로 BOT 객체에서 정의하기 위해 필요한 것이다.

- *0_1_Integer_Variable* ⊂ *Nonnegative_Integer_Variable*
- *Nonnegative_Integer_Variable* ⊂ *Integer_Variable*
- *Constant_1_Coefficient* ⊂ *Nonnegative_Real_Coefficient*
- *Constant_2_Coefficient* ⊂ *Nonnegative_Real_Coefficient*
- *Constant_n_Coefficient* ⊂ *Nonnegative_Real_Coefficient*
- *Nonnegative_Real_Coefficient* ⊂ *Real_Coefficient*
- *Count_Sum_0_1* ⊂ *Volume_Sum_0_1*
- *Volume_Sum_0_1* ⊂ *Volume_Sum_Integer*

3.3 제약식의 Distinctiveness

일반적인 표현형식은 다음과 같다.

ALL BOT

(AT_LHS (Distinctivenesses of BOTs in LHS) ; 좌변항

AND

AT_RHS (Distinctivenesses of BOTs in RHS)) ; 우변항

AND

OPERATOR = XOR [EQ GE LE]

7가지의 전형적인 내포구조를 표현하기 위해 제약식 Chunks 의 Distinctivenesses 는 다음과 같은 10가지로 구분된다.

1) Capacity Constraint

ALL BOT

(AT_LHS (Volume_Sum_0_1)

AND

AT_RHS (nonnegative _Real_ Coefficient))

AND

OPERATOR = LE

2) First_Matching Constraint

ALL BOT

(AT_LHS (Count_Sum_0_1 Sum_with_first_index)

AND

AT_RHS (Constant_1_Coefficient))

AND

OPERATOR = EQ

3) Second_Matching Constraint

ALL BOT

(AT_LHS (Count_Sum_0_1 Sum_with_second_index)

AND

AT_RHS (Constant_1_Coefficient))

AND

OPERATOR = EQ

4) 0_1_GUB (Generalized Upper Bound) Constraint

ALL BOT

(AT_LHS (Count_Sum_0_1 First_XOR_Second_Index_Sum)

AND

AT_RHS (Constant_1_Coefficient))

AND

OPERATOR = EQ

5) 0_1_Bounded Constraint

ALL BOT

(AT_LHS (Count_Sum_0_1)

AND

AT_RHS (Volume_Sum_0_1))

AND

OPERATOR = LE

6) One_Cycle Constraint

ALL BOT

(AT_LHS (Precedence_Node)

AND

AT_RHS (Constant_S-1_Coefficient))

AND

OPERATOR = LE

7) Starting_Node_Degree Constraint

ALL BOT

(AT_LHS (Starting_Node)

AND

AT_RHS (Constant_2_Coefficient))

AND

OPERATOR = EQ

8) Edge_All Constraint

ALL BOT

(AT_LHS (Precedence_Node)

AND

AT_RHS (Constant_n_Coefficient))

AND

OPERATOR = EQ

9) Polynomial_Recursive Constraint

ALL BOT

(AT_LHS (Volume_Sum_Integer Time_Indexed)

AND

AT_RHS (Nonnegative_Real_ Coefficient))

AND

OPERATOR = EQ

10) Group Constraint

ALL BOT

(AT_LHS (Integer_Variable)

AND

AT_RHS (Real_Coefficient))

AND

OPERATOR = LE

3.4 목적함수의 Distinctiveness

일반적인 표현형식은 다음과 같다.

ALL BOT

(OBJECTIVE (Distinctivenesses of the BOTs in Objective Function))

AND

DIRECTION = XOR [MAX MIN]

3.5 내포구조의 Distinctiveness

끝으로, 어느 특정한 정수계획 모형은 제약식과 목적함수의 Distinctiveness 에 의해서 정의된다. 전형적인 7가지의 내포구조를 표현해 보자.

1) 0_1_Knapsack Structure

IF (EXIST CONSTRAINT (capacity))

THEN STRUCTURE = 0_1_Knapsack

2) Assignment Structure

IF EXIST CONSTRAINT (first_matching second_matching))

THEN STRUCTURE = Assignment

3) 0_1_GUB Structure

IF (EXIST CONSTRAINT (0_1_GUB))

THEN STRUCTURE = 0_1_GUB

4) 0_1_VUB Structure

IF (EXIST CONSTRAINT (0_1_ bounded))

THEN STRUCTURE = 0_1_VUB

5) 1_Spanning_Tree Structure

IF (EXIST CONSTRAINT (one_cycle starting_node_degree edge_all))

THEN STRUCTURE = 1_Spanning_Tree

6) Knapsack_with_Time_Horizon Structure

IF (EXIST CONSTRAINT (polynomial_recursive))

THEN STRUCTURE = Knapsack_with_ Time_Horizon

7) General_IP_with_Unbounded Structure

IF (EXIST CONSTRAINT (group))

THEN STRUCTURE = General_IP_with_Unbounded

위의 7가지의 내포구조 들 사이에 나타날 수 있는 여러 Distinctiveness 들을 AND/OR 그래프로 표시하면 Figure 1 과 같다.

Figure 1 appears about here

---

4. 내포구조의 정의

4.1 정수모형의 의미론적 표현

이 절에서는 정수계획 문제를 의미론적인 관점에서 표현하고자 한다. 아래의 문제는 plant assignment problem 이다. 이것을 UNIK-IP 도구를 이용하여 표현하고, 또한 그것을 가지고 추론과정을 거쳐 내포구조를 정의하고자 한다.

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

subject to

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

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

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

여기서 i 와 j는 product와 plant를 의미한다.

위의 문제는 generalized assignment problems 의 한 종류이다. Figure 2 는 위의 문제를 표현한 것이며, 변수와 계수 객체의 Distinctiveness 가 나타나 있지 않다. 그러나 라그란지안 완화를 적용하기 위해 모든 각 객체의 Distinctiveness 를 정의하지 않으면 안된다. UNIK-RELAX 는 데이타베이스에서 데이타의 특성을 검색하여 계수 객체의 Distinctiveness 를 정의한다. 특히, 변수는 TYPE 슬롯의 값이 'binary'로 윗식의 (9)를 나타낸다.

Figure 2 appears about here

4.2 상향식 추론방식

상향식 추론방식은 변수 및 계수 객체를 먼저 정의하고, BOTs, 제약식 객체, 그리고 모형의 내포구조를 정의한다. UNIK-RELAX는 Figure 2의 plant assignment problem 으로 부터 *'distinctiveness'* 와 *'embedded_structure'* 를 정의한다 이것을 Figure 3에서 잘 보여주고 있다. 즉 정수계획 모형 plant assignment problem 은 두개의 내포구조를 갖게된다. 그것은 0_1_Knapsack 과 0_1_GUB 내포구조이다.

Figure 3 appears about here

상향식 추론방식에서 두개 이상의 내포구조가 발견되면 Table 2 [1, 2, 9]에서 나타난 내포구조간의 효율성에 따라 정리된 것을 이용하여 Conflict를 해결한다. 위의 표에서 보듯이 0_1_GUB 보다 0_1_Knapsack이 우선순위가 높음을 알 수 있다. Table 2는 내포구조간의 우선순위를 나타내고 있으며, 이는 실험적인 결과를 통해서 요약 정리되었다.

Table 2 appears about here

이것을 실현하기 위해 전방향 추론도구 (UNIK-FWD) 를 이용하였으며, 관련되는 규칙은 Figure 4 에서 설명하고 있다.

Figure 4 appears about here

4.3 하향식 추론방식

하향식 추론방식은 Table 2에서 나타난 효율성 우선순위에 따라 정수계획모형의 내포구조를 정의해 나가는 방식이다. 미리 정의하고자 하는 내포구조를 가지고 하향식의 방향으로 각 객체 별로 정의하게 된다. UNIK-RELAX 에서는 이 추론방식을 역방향 추론도구인 UNIK-BWD 와 같이 이뤄지며, Figure 5 는 UNIK-BWD 문법에 맞게 작성된 것이며, 0_1_Knapsack 구조를 정의하는 규칙이다.

Figure 5 appears about here

4.4 경험적 추론방식

제 2 장에서는 generalized assignment problem 와 같은 Np-hard 인 문제가 효과적으로 라그란지안 완화가 가능하다고 언급하였다. 이러한 Np-hard 문제와 라그란지안 문제로 연결되는 관계들이 있으면 그것을 참조하여 처음에 모형을 정의만 하면 해당되는 라그란지안 모형은 쉽게 얻을 수 있음을 알 수 있다 (Fisher, 1981).

Table 3 appears about here

4.5 추론방식들의 통합

위에서 언급한 3가지 방식 - 상향식 추론방식, 하향식 추론방식, 경험적 추론방식 - 을 종합하면 Figure 6 과 같다.

Figure 6 appears about here

처음의 정수계획 모형에 대한 구조를 미리 알 수 있으면 경험적 추론방식을 적용하는 것이 효과적이며, 그렇지 않고 내포구조에 대한 부분적인 정보가 주어졌다면 하향식 추론방식이 좋다. 아무 정보가 주어지지 않았을 경우는 상향식 추론방식에 따라 각 객체를 밑에서 위로 정의하여야만 한다.

5. 라그란지안 문제로의 완화

앞에서 내포구조가 정의되면 그 내포구조를 가지고 Table 1에 따라 라그란지안 문제가 정의되고, 의미론적 표현 상에서의 라그란지안 완화 절차가 이루어진다. 그 완화 절차에는 5가지 있다.

Step 1. 완화될 제약식 객체를 파악한다.

- 만약에 내포구조가 Table 1 의 구조 1-5 중에 하나이면, 구조와 연관이 없는 제약식을 선택한다.
- 만약에 내포구조가 Table 1 의 구조 6 이면, 구조와 연관된 제약식을 선택한다.
- 만약에 내포구조가 Table 1 의 구조 7 이면, unbounded variables 과 연관된 제약식을 선택한다.

[Example]

plant assignment 문제는 Table 1 의 구조 1 *(i.e. 0_1_Knapsack)* 을 갖는다.

0_1_Knapsack 과 연관이 없는 *plant_assignment_constraint* 를 완화한다.

Step 2. 람다 객체의 라그란지안 승수를 위한 새로운 속성을 생성한다.

[Example]

람다 객체는 Figure 7 의 (4)에 나타나 있다. 람다 계수의 생성을 위해서는 다음의 규칙을 적용한다[12].

IF *OPERATOR* of relaxed constraint is EQ,

THEN *DISTINCTIVENESS* of lambda object is real.

IF *OPERATOR* of relaxed constraint is LE,

THEN *DISTINCTIVENESS* of lambda object is nonnegative.

IF *OPERATOR* of relaxed constraint is GE,

THEN *DISTINCTIVENESS* of lambda object is nonpositive.

Step 3. 계수의 라그란지안 승수를 포함하는 새로운 BOT 들을 생성한다.

[Example]

새로운 BOTs 들은 Figure 7 의 (2) 와 (3) 에 있다.

Step 4. IP-model 객체의 OBJECTIVE 에 라그란지안 BOT 들을 추가한다.

[Example]

추가된 BOT 들은 Figure 7 의 (1) 에 있다.

Step 5. Step 1 에서 파악된 완화될 제약식들과 이들과만 연관된 BOT 들을 삭제한다.

[Example]

이 예에서 제약식 *plant_assignment_constraint* 와 *plant_choice* BOT 와 one BOT 가 Figure 3 에서 삭제되었다.

6. UNIK-RELAX의 설명

내포구조의 자동적 인식과 라그란지안 문제의 생성을 구현하기 위해서 우리는 UNIK-RELAX. 시스템을 개발하였다. UNIK-RELAX 의 전체구조는 Figure 8 에 나타나 있다. UNIK-RELAX의 주요 화면을 설명하면 다음과 같다.

Figure 8 appears about here

---

(1) UNIK-IP로 부터 정수계획 모형 입력 은 Figure 9와 같다.

Figure 9 appears about here

---

(2) 내포구조 정의 결과 화면은 Figure 10과 같다.

Figure 10 appears about here

---

(3) 완화된 라그란지안 모형을 나타낸 화면은 Figure 11과 같다.

Figure 11 appears about here

---

7. 결론

내포구조의 자동적 인식과 라그란지안 문제의 생성을 구현하기 위해서 우리는 UNIK-RELAX. 시스템을 개발하였다. UNIK-RELAX 의 전체구조는 Figure 10 에 나타나 있다. 전절에서 설명한 주요한 모듈 - Structural Identification 와 Lagrangian Problem Generation - 이외에도 Solver Controller 는 라그란지안 승수를 위한 적절한 숫자를 생성하고 primal solution algorithm (Branch and Bound) 을 라그란지안 문제에 의해서 계산된 한계와 연계한다.

UNIK-RELAX 는 UNIK 개발도구를 이용해서 개발되었다. 특히, 객체표현 (UNIK-OBJECT), 정방향추론 (UNIK-FWD) 과 역방향추론 UNIK-BWD), 그리고 선형 (UNIK-LP) 과 정수계획모형 (UNIK-IP) 표현과 모형생성 지원 기능등을 활용하였다[23].

이 연구는 계산적으로 복잡한 정수계획 모형에서의 전형적인 내포구조를 자동적으로 인식하고, 완화된 라그란지안 문제로의 자동적인 변환이 가능함을 보여주었다.

감사의 글 : 이 연구를 박사과정 시절에 논문을 지도해주시고, 'the world of complexity'를 가르쳐 주셨던 나의 가장 존경하는 Marshall L. Fisher 교수님에게 받칩니다. (이재규)

References

[1] Aggarwal, V., A Lagrangian Relaxation Method for the Constrained Assignment Problem, Comput. & Ops. Res., Vol 12, (1995).

[2] Balas, E. and N. Christofides, Talk presented at the Ninth International Synposium on Mathematical Programming, Budapest, August, (1976).

[3] Bazaraa, M. S. and J. J. Goode, The Traveling Salesman Problem: A Duality Approach, Math. Programming, Vol 13 (1977)

[4] Burdet, C. A. and E.L. Johnson, A Subadditive Approach to solve linear Integer Programs, Ann. Discrete Math., Vol. 1(1977).

[5] Chalmet, L. G. and L. F. Gelders, Lagrangian Relaxation for Generalized Assignment Problem, Proc. Second European Congress on Operations Research, North-Holland, Amsterdam (1976)

[6] Cornuejols, G., M. L. Fisher, and G. L. Nemhauser, Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms, Management Sci., Vol. 23 (1977).

[7] Etcheberry, J., The Set-Covering Problem: A New Implicit Enumeration Algorithm, Operations Research, Vol 21 (1977).

[8] Etcheberry, J., C. Conca and E. Stacchetti, An Implicit Enumeration Approach for Integer Programming Using Subgradient Optimization, Pub, No. 78/04/c, Universidad de Chile, (March 1978)

[9] Fisher, M. L., The Lagrangean relaxation method for solving integer programming problems. Management Science, Vol. 27, No. 1, (Jan 1981).

[10] Fisher, M. L. and J.F. Shapiro, Constructive Duality in Integer programming. SIAM J. Apple. Math., Vol 27 (1974).

[11] Fisher, M. L., A Dual Algorithm for the One-Machine Scheduling Problem, Math. Programming, Vol 11 (1976)

[12] Fisher, M. L., Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I, Operations Res., Vol. 21 (1973)

[13] Fisher, M. L. and D. S. Hochbaum, Database Location in Computer Networks, J. Assoc. Comput. Mach, (1981).

[14] Fisher, M. L., G. L. Nemhauser, and L. A. Wolsey, An Analysis of Approximation for Finding a Maximum Weight Hamiltonian Circuit, Operations Res., Vol. 27 (1979)

[15] Fisher, M. L., R. Jaikumar, and L. Wassenhove, A Multiplier Adjustment Method for the Generalized Assignment Problem, Decision Sciences Working Paper, University of Pennsylvania (May 1980)

[16] Fong, C. O. and M. R. Rao, Capacity Expansion with two producing regions and concave costs, Management Science, 22 (1975)

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

[18] Geoffrion, A. M. and R. Mcbride, Lagragian Relaxation Applied to Capacitated Facility Location Problems, AIIE Trans., Vol. 10 (1978)

[19] Held, M. and R. M. Karp, The Traveling-Salesman problem and Minimum Spanning trees, Operations Res. Vol 18 (1970)

[20] Kim, Chulsoo, Jae K. Lee, and M. Kim, Automatic Model Structural Identification for Lagrangian Relaxation : UNIK-RELAX, Proceedings of the 3rd The International Society for Decision Support Systems, 95-105, (1995).

[21] Kim, Wooju and Jae K. Lee, UNIK-OPT/NN: Neural Network-based Adaptive Optimal Controller on the Optimization Models, forthcoming in Decision Support Systems, (1995).

[22] Krishnan, R., PDM: A Knowledge-based tools for Model Construction. Proceedings of the 22nd Hawaii International Conference on System Science, 3, 467-474, (1989)

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

[24] Lee, Jae K., Integration and Competition of AI with Quantitative Methods for Decision Support, Expert Systems With Applications 1, (1990).

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

[26] Lee, Jae K. and S.B. Kwon, ES* : An Expert Systems Development Planner Using A Constraint and Rule-Based Approach, Expert Systems with Applications 9, No. 1, (1995).

[27] Lee, Jae K. and Y.U. Song, UNIK-PMA: A Unifier of Optimization Model with Rule-Based Systems by Post-Model Analysis, forthcoming in Intelligent Systems in Accounting, Finance, and Management, (1995).

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

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

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

[31] 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)

[32] Murphy, F. H., E.A. Stohr, and P. Ma, Composite Rules For Building Linear Programming Models From Component Models, Management Science 38, No. 7, (1992).

[33] Nemhauser, G. L. and G. Weber, Optimal Set Partitioning Matchings and Lagrangian Duality, ORSA/TIMS Meeting, (1978).

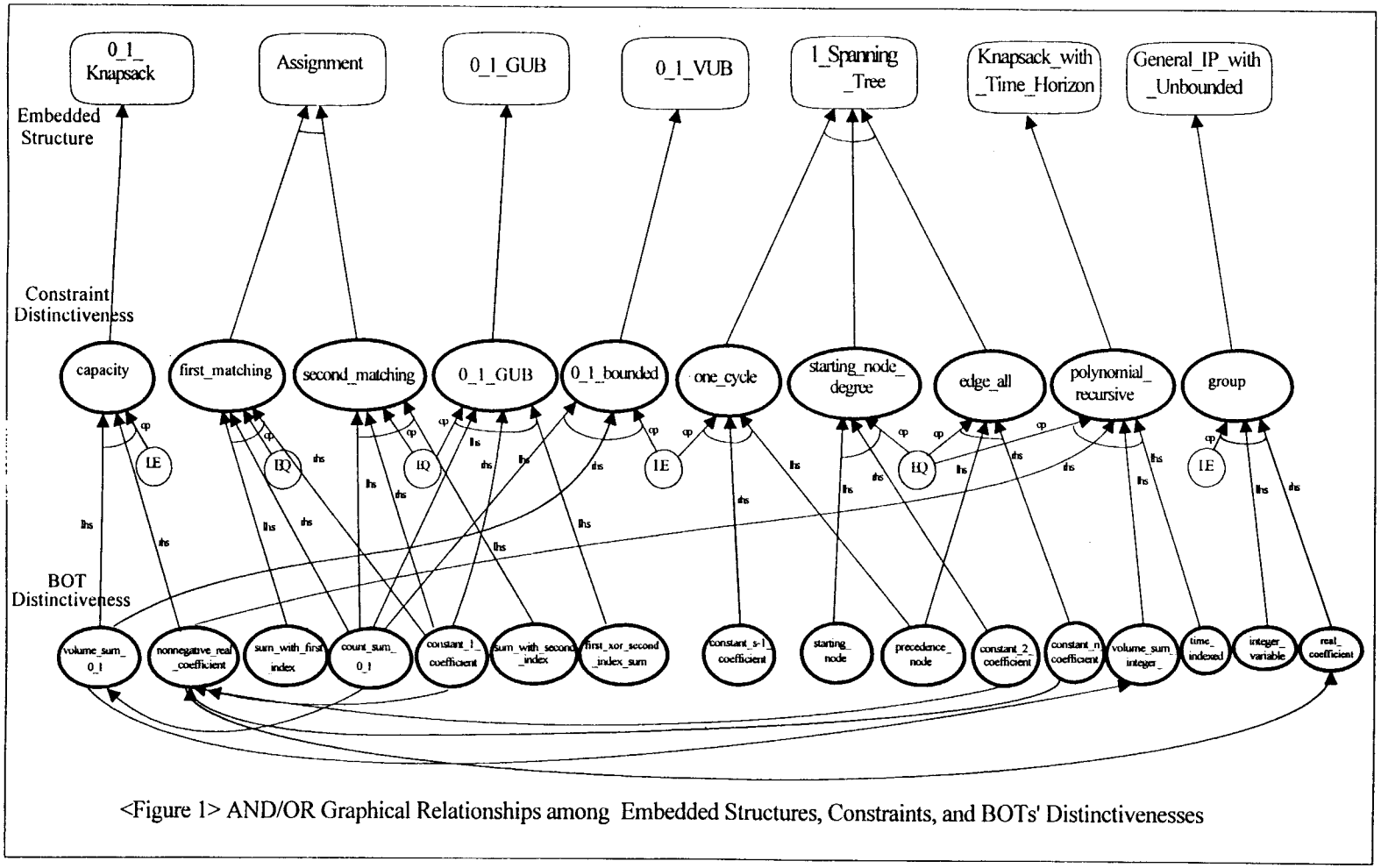
[34] Yeom, Kuhn and Jae K. Lee, Knowledge-Assisted Integer Programming Formulator : UNIK-IP, Proceedings of INFORMS Los Angeles Spring 1995 National Meeting, (1995).

<Table 2> Efficiency Priority Between Embedded Structures

	0_1_Knapsack	Assignment	0_1_GUB	0_1_VUB	1_Spanning _Tree
0_1_Knapsack					
Assignment	Assignment				
0_1_GUB	0_1_Knapsack	Assignment			
0_1_VUB	0_1_VUB	Assignment	0_1_VUB		
1_Spanning _Tree	1_Spanning _Tree	1_Spanning _ tree	1_Spanning _Tree	1_Spanning _Tree	

<Table 3> Types of Integer Programming Problems and Adequate Lagrangian Problems

Types of IP Problems		Adequate Lagrangian Problems	Researchers
Generalized Assignment		0-1 Knapsack	Chalmet et al.(1976)[5] Fisher et al.(1980)[15]
General IP	- Unbounded Variables	Group Problem	Fisher & Shapiro(1974)[10] Burdet(1977)[4]
	- 0-1 Variables	Generalized Upper Bound	Etcheberry(1978) [8]
Set Covering and Partitioning	- Covering	Generalized Upper Bound	Etcheberry(1977) [7]
	- Partitioning	Matching	Nemhauser and Weber(1979) [6]
Location	- Uncapacitated	Variable Upper Bound	Cornuejols, Fisher, and Nemhauser (1977) [6]
	- Capacitated	Variable Upper Bound	Geoffrion(1978)[18]
	- Databases in Computer Network	Variable Upper Bound	Fisher et al. (1981) [9]
Scheduling	- n|m weighted Tardiness	Pseudo Polynomial DP	Fisher (1973) [12]
	- 1 Machine Tardiness	Pseudo Polynomial DP	Fisher (1976) [11]
	- Unit Commitment	Pseudo Polynomial DP	Muckstadt (1977) [30]
Traveling Salesman	- Symmetric	Spanning Tree Perfect 2-Matching	Held et al.(1970) [19] Balas et al (1976) [2]
	- Asymmetric	Spanning Tree Assignment	Bazaraa et al.(1977) [3] Balas et al.(1976) [2]



<Figure 1> AND/OR Graphical Relationships among Embedded Structures, Constraints, and BOTs' Distinctivenesses

```
{{ plant_assignment_problem                    ;This model is an example of
      IS-A : IP_MODEL                          ; Generalized Assignment Problem
      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      ; pair of coefficient and variable composed a BOT
      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                ; a coefficient composed a BOT
      SUMMATION_INDEX : }}
{{ plant_choice_BOT
      IS-A : BOT
      ATTRIBUTE : one assignment_var
      SUMMATION_INDEX : plant }}
{{ one_BOT
      IS-A : BOT
      ATTRIBUTE : one                           ; a constant
      SUMMATION_INDEX : }}

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

{{ plant_capacity
      IS-A :COEFFICIENT
      SYMBOL : b
      LINKED_INDEX : plant}}

{{ product
      IS-A : INDEX
      SYMBOL : i
      LINKED_ATTRIBUTE : assignment_var
        unit_cost plant_volume }}

{{ plant_volume
      IS-A : COEFFICIENT
      SYMBOL : a
      LINKED_INDEX : product plant }}

{{ one
      IS-A :COEFFICIENT
      SYMBOL : 1
      LINKED_INDEX : }}

{{ plant
      IS-A : INDEX
      SYMBOL : j
      LINKED_ATTRIBUTE : assignment_var
        unit_cost plant_volume plant_capacity }}
```

<Figure 2> An Illustrative Formulation of Plant Assignment Problem Using UNIK-IP

```
{{ 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  0_1_GUB }}

{{ 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}}

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

{{ plant_sum_BOT
    IS-A : BOT
    ATTRIBUTE : plant_volume  assignment_var
    SUMMATION_INDEX :  product
    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 }}

{{ 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 }}

{{ plant_volume
    IS-A : COEFFICIENT
    SYMBOL : a
    LINKED_INDEX : product plant
    DISTINCTIVENESS : nonnegative }}

{{ plant_capacity
    IS-A : COEFFICIENT
    SYMBOL : b
    LINKED_INDEX : plant
    DISTINCTIVENESS :nonnegative  }}

{{ one
    IS-A : COEFFICIENT
    SYMBOL : 1
    LINKED_INDEX :
    DISTINCTIVENESS : constant }}

{{ product
    IS-A : INDEX
    SYMBOL : i
    LINKED_ATTRIBUTE : assignment_var
        unit_cost  plant_volume  }}

{{ plant
    IS-A : INDEX
    SYMBOL : j
    LINKED_ATTRIBUTE : assignment_var
        unit_cost  plant_volume  plant_capacity }}
```

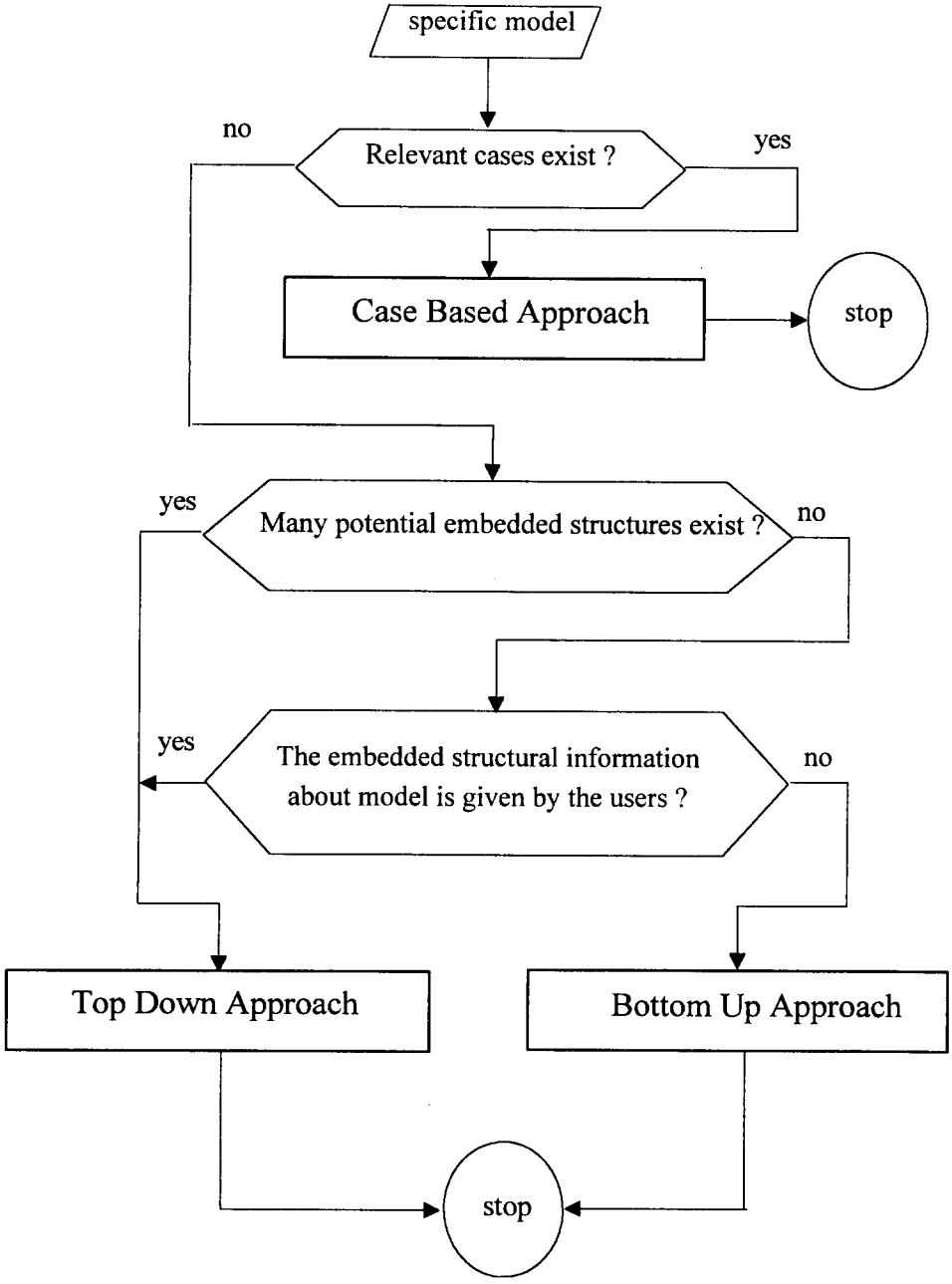
<Figure 3> Identified Embedded Structures in the Plant Assignment Problem

```
(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  ))
```

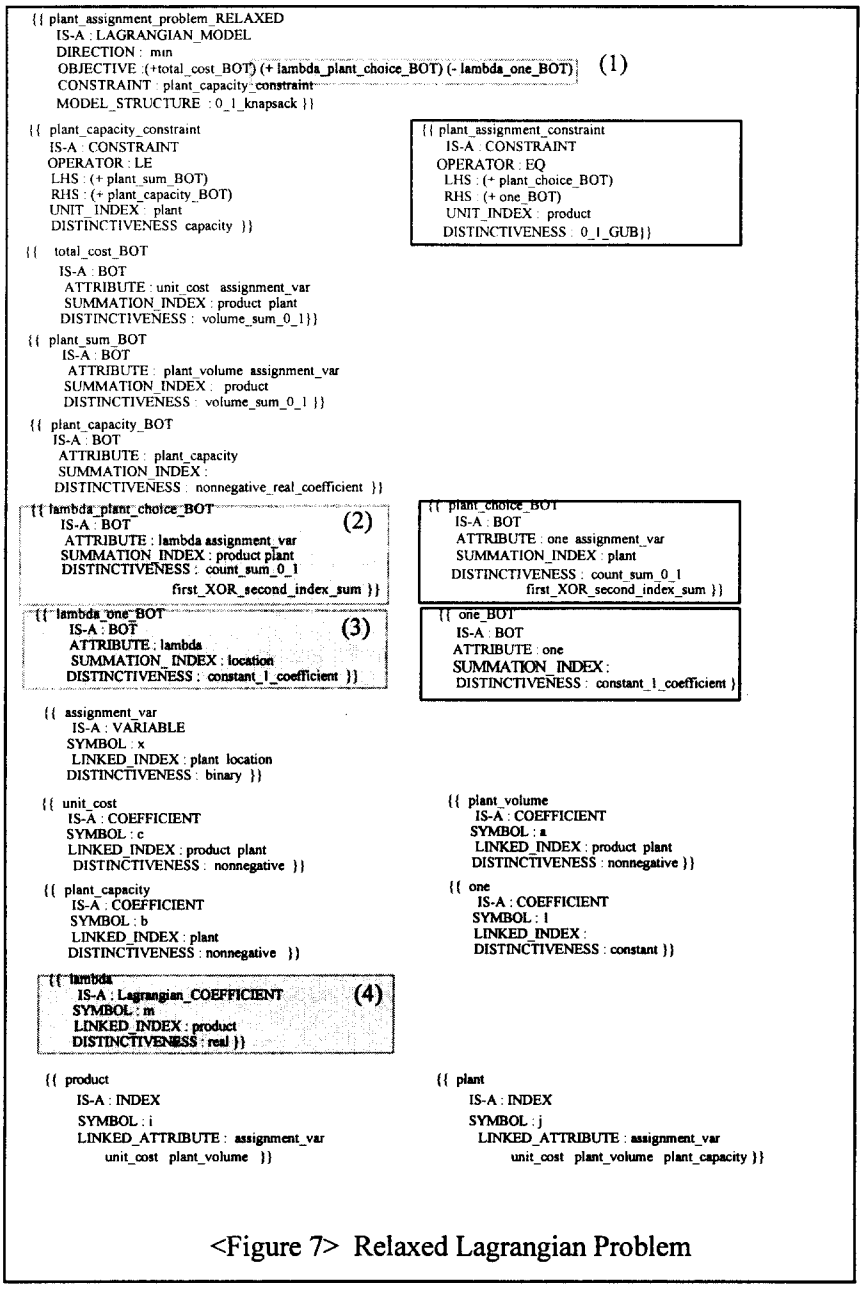
<Figure 4> Illustrative Rules for the Identification of 0_1_Knapsack Structure in the UNIK-FWD Syntax

```
(RULE 0_1_knapsack_structure
[RULE_GROUP MODEL]                                  ; meta-rule about rule group
[PRIORITY 1 ]                                       ; meta-rule about priority
IF    (IS CONSTRAINT.DISTINCTIVENESS  'capacity )
THEN (IS EMBEDDED_STRUCTURE  '0_1_knapsack))

(RULE capacity_constraint
[RULE_GROUP CONSTRAINT]
IF     (IS  BOT.DISTINCTIVENESS OF RHS   'nonnegative_real_coefficient )
AND  (IS  BOT.DISTINCTIVENESS OF LHS  'volume_sum_0_1 )
AND  (IS  OPERATOR  'LE )
THEN (IS  DISTINCTIVENESS  'capacity ))

(RULE volume_sum_0_1_BOT
[RULE_GROUP BOT]
IF   (IS  ATTRIBUTE.DISTINCTIVENESS OF ATTRIBUTE   ('nonnegative_real_coefficient  '0_1_integer_variable))
THEN (IS  DISTINCTIVENESS 'volume_sum_0_1 ))

(RULE  nonnegative_real_coefficient_BOT
[RULE_GROUP BOT]
IF      (IS  ATTRIBUTE.DISTINCTIVENESS  OF ATTRIBUTE  'nonnegative )
THEN (IS  DISTINCTIVENESS  'nonnegative_real_coefficient ))
```

<Figure 5> Illustrative Rules associated with 0_1_Knapsack Structure in the UNIK-BWD Syntax



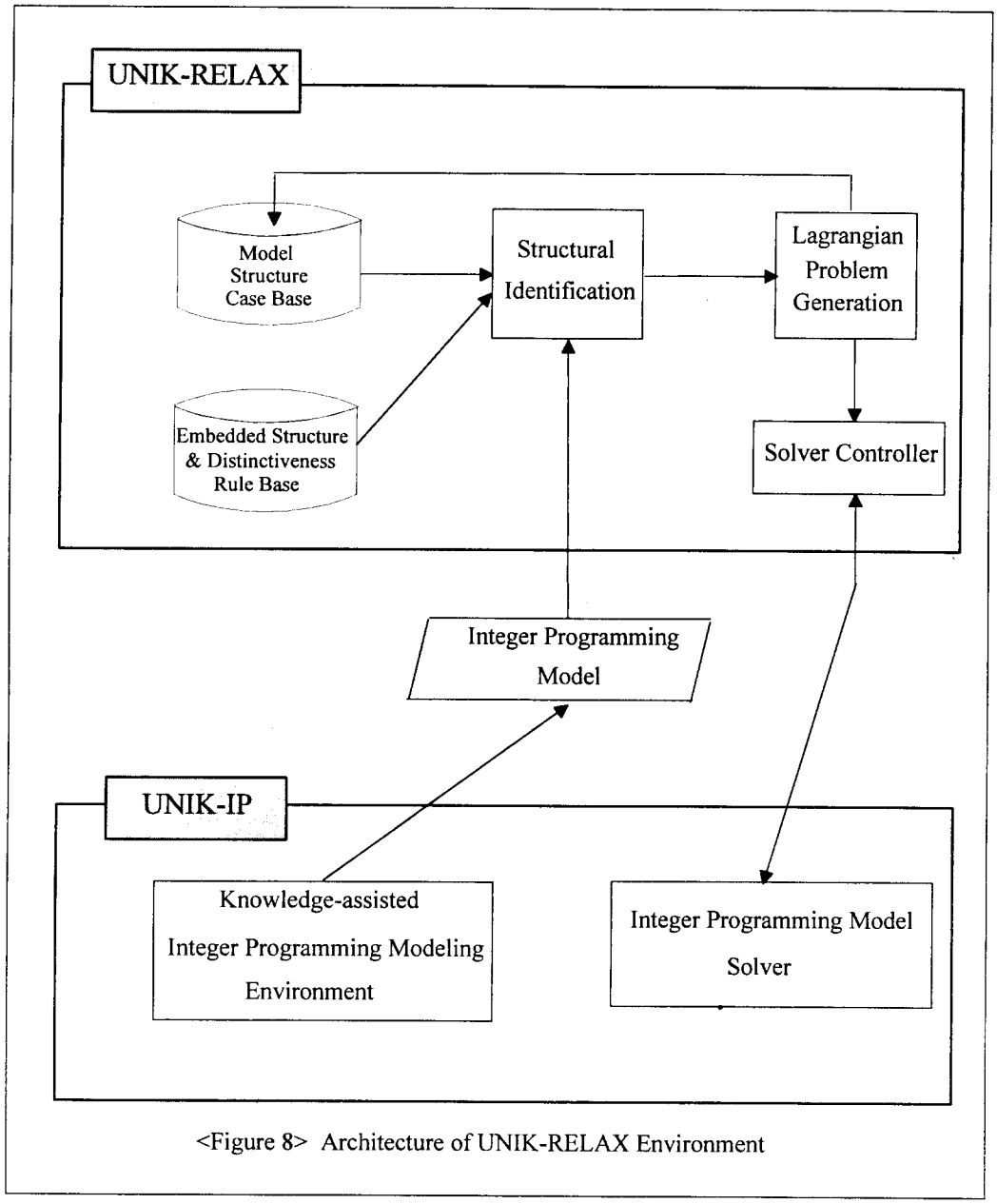
<Figure 6> Integration of three approaches



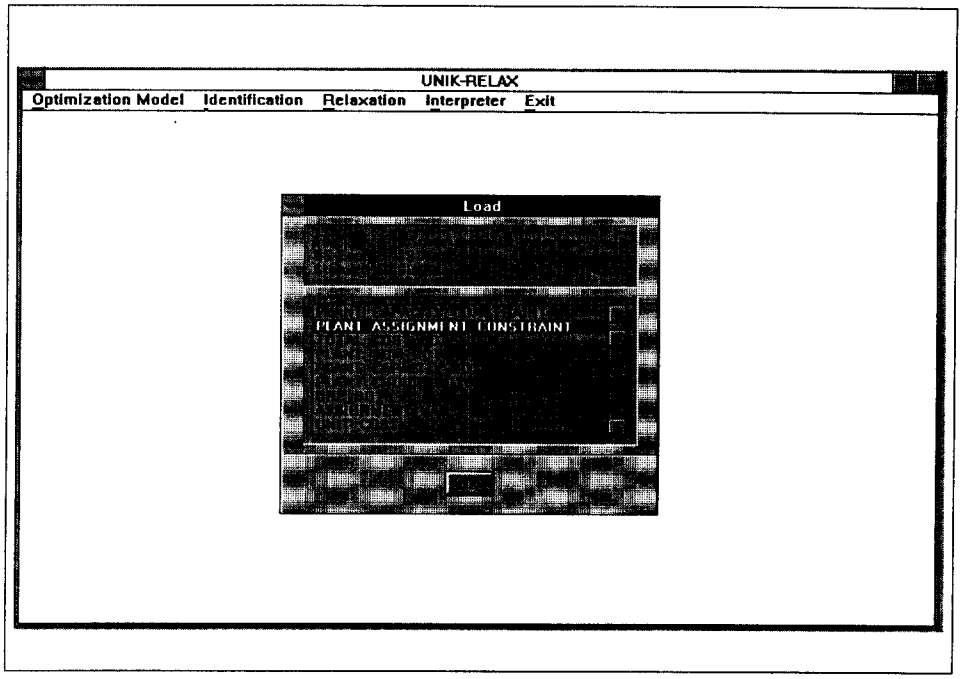
<Figure 7> Relaxed Lagrangian Problem

. Generated Objects

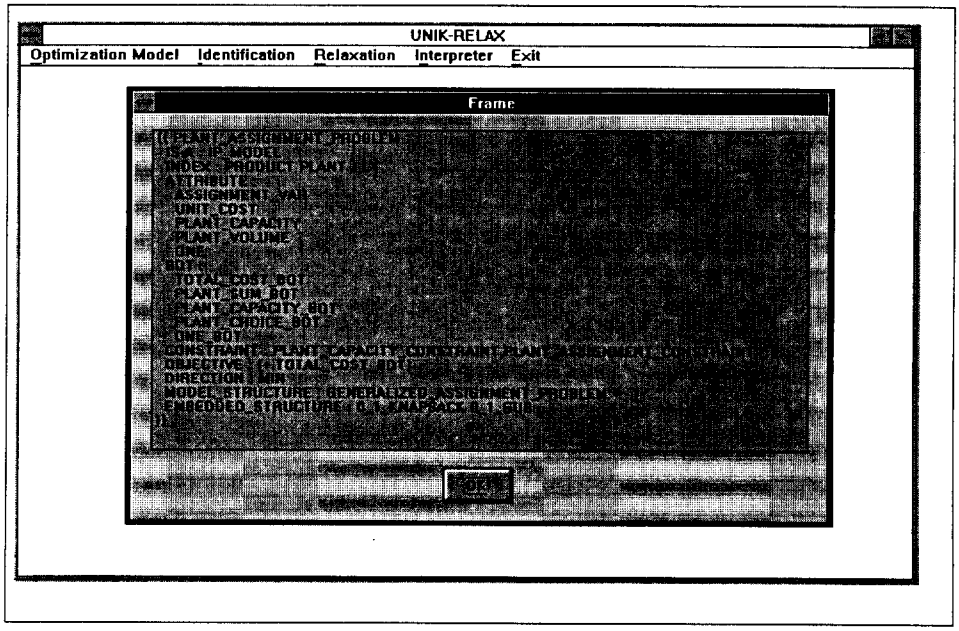
. Eliminated Objects



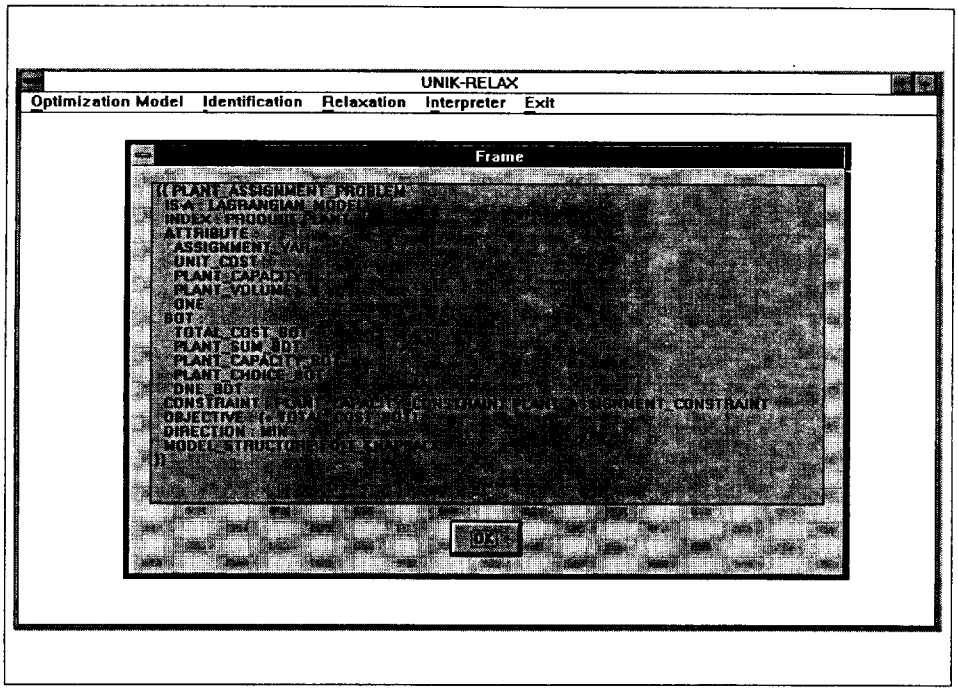
<Figure 8> Architecture of UNIK-RELAX Environment



<Figure 9> Integer Programming Input Screen



<Figure 10> Structural Identification Resulting Screen from IP Model



<Figure 11> Lagrangian Problem Generation Resulting Screen