

규칙과 논리프로그래밍을 이용한 탄력적 제약 만족문제 해법

김 철 수[†]

요 약

본 연구에서 제시한 구조는 다중목적은 갖는 규칙기반 제약만족문제의 표현을 포괄하고 있고 그 문제는 시스템에서 변화에 탄력적으로 대응할 수 있는 전력공급계획을 효과적으로 수립한다. 이 연구에서 제안한 구조를 계량적 모델을 위한 모델링 시스템과 정성적 모델을 위한 규칙 기반의 제약만족 프로그래밍 기법을 이용하였다. 또한 좋은 사례를 통한 실험결과를 요약하면 이 문제의 표현을 포함한 구조는 효과적이며 탄력적인 의사결정을 얻는데 큰 도움을 주며, 발전기 특성을 고려하는 데에서는 기존의 문제 구조보다는 표현이 다양하고 쉽다는 장점을 갖는다.

A Method for Reactive Scheduling using Rule-based Constraint Satisfaction Problem

Chul-Soo Kim[†]

ABSTRACT

A framework proposed in this paper includes representations of multi-objectives rule-based constraint satisfaction problem(RCSP) and especially, flexible structure of RCSP which provides adequate solution within an expectation time and adaptively considers some change of constraints on the RCSP. In this study, the power system implemented by above framework is entirely programmed by using a modeling system for the quantitative model and a rule-based constraint programming tool for the qualitative model. The results of an exemplary case show how the structure that involves the problem representation gives the scheduler an effective decision and what are the advantages of the easy and diverse representation for considering the technical characteristics of the units on the RCSP over the traditional problem.

1. 서 론

기업의 의사결정문제는 상위단계로 갈수록 다양해지며 복잡해지는 경향을 보이고 있다. 해를 구하는 측면에서 의사결정 문제의 구조가 복잡해지고 제약식이 많아지게 되면 최적해를 찾기란 여간 어렵지가 않다. 또한 그러한 문제에서는 제한적인 제약과 정성적인 제약

이 혼합되어있는 경우가 대부분이다. 모델링이나 해법도 계량적(Quantitative)인 접근법인 최적화 기법(Optimization Method)과 정성적(Qualitative)인 제약 해결 기법인 제약만족(Constraint Satisfaction) 해결기법을 같이 고려해야만 풀 수 있는 문제구조를 가지고 있다. 본 논문이 연구대상으로 삼고 있는 전력공급계획 문제도 그와 같은 문제구조를 지니고 있다.

전력발전시스템에서 예기치 않은 전력수요의 상승이나 발전기의 고장 등은 전력공급계획을 수립하는 데에 매우 어려움을 더해준다. 전력공급에 참여하는 발전기

※ 본 논문은 2000년도 인허대학교의 지원에 의해 수행되었음(Inha-21090).

† 정 회 원 . 인허대학교 경영학부 교수
논문집수 · 2000년 2월 11일, 심사완료 · 2000년 9월 12일

들의 기계적인 특성을 고려하는 것은 전력공급계획문제에서 중요한 요소이며, 그 계획문제는 계량적인 제약과 정성적인 제약하에서 경제적 전력공급과 발전기의 안정성 제고의 두 가지 목적을 동시에 충족하여야 하는 문제로 설명된다. 그러나 발전기의 기계적인 문제를 고려하는 정성적인 제약식이 매우 복잡한 방법으로 문제에 섞여 있어서 그 제약식을 고려하는 통합문제는 많은 어려움을 갖고 있다. 이 연구에서는 지식기반 탄력적 전력공급계획 구조를 제안한다.

본 연구에서 제시한 구조는 다중목적성을 갖는 규칙기반 제약만족문제의 표현을 포함하고 있고 그 문제는 시스템에서 변화에 탄력적으로 대응할 수 있는 전력공급계획을 효과적으로 수립한다. 효과적이고 탄력적인 공급계획을 수립하기 위해서는 인공지능(AI)과 계량경영(OR)에서 활용되는 이론적인 방법들을 통합한 접근법을 이용하는 것이 유리한데 그 통합적 접근법을 프로그래밍하기 위해서 계량적 모델을 표현하기 위해서 모델링 시스템을 이용하였으며[6] 정성적 모델 표현을 위해서 규칙기반 중심으로 제약만족문제(CSP) 프로그래밍 도구를 활용하였다. 또한 좋은 사례를 통한 실험 결과를 요약하면 이 문제의 표현을 포함한 구조는 효과적이며 탄력적인 의사결정을 얻는데 큰 도움을 주며 발전기의 기계적 특성을 고려하는 데에서는 기존의 문제 구조보다는 표현이 다양하고 쉽다는 장점을 갖는다

2. 지식기반시스템과 논리 프로그래밍

지식기반시스템은 모델관리 분야에서 큰 역할을 보이며 주고 있으며 규칙을 포함한 이러한 지식기반시스템의 역할은 세 가지로 구분될 수 있는데, 모델링 이전의 지식기반시스템, 모델링 이후의 지식기반시스템, 그

리고 모델과 해법사이에서의 지식기반시스템으로 구분할 수 있다. 위의 세 가지 관점에서 지식기반시스템의 역할과 그 시스템에서 이용한 규칙을 설명하면 다음과 같다.

첫째로, 모델링 이전의 지식기반시스템에 관한 연구로는 개념적 모델 SM(Structured Model : SM)을 들 수 있다[1] 둘째로, 모델링 이후의 지식기반시스템에 관한 연구는 최적화 모델을 풀고 난 후의 해가 지식기반시스템내의 데이터베이스에 입력되어 재무정보로 해석하는 연구가 있다. 여기서는 다양한 지식이 제시되고 있다[3]. 끝으로 모델과 해법사이에서의 지식기반시스템에 관한 연구로는 객체형 중심의 데이터베이스를 중심으로 모델베이스를 구축하여 해법에 직접 연결시켜 주는 연구가 있다

제약만족 프로그래밍은 두 가지 측면에서 생각할 수 있다. 하나는 수학적 논리와 자동 가설증명에 관련된 것이고, 다른 하나는 논리에 근거한 프로그래밍 언어의 사용과 개발에 관련된 것이다. 제약만족 프로그래밍 언어(Logic Programming Language)는 다양한 문제를 해결하는 분야에서 그 능력을 발휘해 왔고, 많은 응용분야에서 심볼릭(Symbolic)의 수식의 문제와 같은 영역에 있어서도 논리 언어의 적용을 활발하게 적용하는 단계에 이르렀다 그래픽 계산에 대한 논리 언어의 적용을 시도하고 있으며, 그래픽 계산에 대한 논리 언어의 요구가 증가하고 있는 점을 고려할 때에는 수학적 논리 표현의 한계를 극복하는 다양한 서도가 요구되고 있다.

제약 논리프로그래밍에서의 제약 해결기는 OR, 경영과학, 산업공학에서 주로 사용하는 심플렉스법의 변형을 CLP(R), Prolog III와 같은 언어에서 이용하였으며,

<표 1> 제약만족문제 해법기 비교

	CLP(R)	CHIP	Prolog III
개발자	IBM T.W.Reasrch	European Computer-Industry Research Center	Alan Colmerauer and his group
제약 해결기	The Simplex variant	CSP Technique for Finite Domain optimization The Simplex for rationals Local Propagation for Boolean	The Simplex variant
도메인	Real	Discrete Finite Domain(Integer) Rationals Boolean	Rationals Boolean
처리 가능한 제약	=, <, >, ≤, ≥	=, ≠, <, >, ≤, ≥	=, ≠, <, >, ≤, ≥

CHIP에서는 제약만족문제 개념을 논리 프로그래밍에 통합하여 사용하였다. 즉, CLP(R)과 Prolog III와 같은 심플렉스 기반의 제약해결 기법을 사용하는 언어들은 제약 해결기가 언어의 설계차원에서 논리언어로 입력된다[2].

3. 제약만족문제의 구조

제약만족문제에서 제약은 일반적으로 의사결정과정에서 유지되어야 하는 관계와 함수들로서 정성적 제약과 계량적 제약이 있다. 그래서 정성적이고 계량적인 제약만족문제(Qualitative/Quantitative Constraint Satisfaction Problem: QQCSP)는 하나의 실험적인 관계형 시스템(Empirical Relational System) W 로 개념화될 수 있는데, 그 시스템은 고려대상이 되는 객체, 관계, 함수 등으로 구성된다[5].

QQCSP의 모델-이론적인 관점은 세 가지 단계로 구분되는데,

첫째는 $W = \{O, R, F\}$ 로 표시되는 개념적 층(Conceptual Level)이다. 여기서 O 는 객체(Object)들의 집합이며, R 은 관계(Relation)들의 집합을 말하며, F 는 함수(Function)들의 집합이다.

둘째는 표현적 층(Representational Level)으로 W 를 정형적으로 표현한다. 여기서는 제약논리프로그래밍(CLP) 접근법은 하나의 논리모형(Logic Model)과 하나의 수리모형(Mathematical Model)인 정형화된 모델들을 갖는다. 이것을 나타내면 다음과 같다. $W = \{E, R_E, F_E\} \cup \{N, R_N, F_N\} \cup \{EUN, R_{EN}, F_{EN}\}$. 여기서 E 는 심볼객체(Symbolic Object)들의 집합, R_E 는 E 중에서 관계들의 집합, F_E 는 E 중에서 함수들의 집합이다. 그리고 N 은 실수의 집합, R_N 은 N 중에서 관계들의 집합, F_N 은 N 중에서 함수들의 집합을 의미하며 R_{EN} 은 E 와 N 중에서 관계들의 집합을 뜻한다. 이러한 수리적인 제약과 논리적인 제약이 서로 존재하는 경우에는 독립적으로 풀 수가 없다. 즉, $\{EUN, R_{EN}, F_{EN}\}$ 은 두 제약이 의존적인 관계를 갖게 된다.

셋째는 계산적 층(Computational Level)으로 표현적 층의 구현으로 볼 수 있다. 여기서는 표현적 층이 두 가지의 정형화된 표현으로 이루어졌기 때문에 두 가지

의 해를 찾는 접근법도 두 가지로 존재한다. 즉, 논리적 추론기법과 수학적 계산법이다. CLP의 중요한 특징은 두 가지의 접근법이 제약을 푸는 과정에서 하나의 접근법이 다른 접근법에 내부 연결되어 있다는 점이다. 즉, 구문론적인 통합이 실패했다든지 계량적 제약이 비가능해(Infeasible)인 경우에는 그 인더프리티는 역추적(Backtracking)을 하게 되는데, 이러한 징성적이고 계량적인 제약이 독립적으로 풀리지 않지만 동시에 만족되어야 하기 때문에 이런 QQCSP 문제는 시스템적 접근법이 필요하게 된다.

이 장에서는 제약만족문제의 한 예로서 발전시스템에서 중요한 의사결정문제인 전력생산계획을 설명하기로 한다. 전력생산계획은 발전 시스템의 단기 운영계획이며, 이 계획을 수립하는 문제는 일정기간동안 시간대 별로 킷두 부하와 비킷두 부하 사이에 심한 변동을 보이는 전력수요를 만족시키면서, 총 발전 비용을 최소화하는 전력생산계획 문제이다. 이 문제는 두 가지 차원의 의사결정문제로 구분하는데, 첫째인 전력생산에 참여하는 발전기를 구성하는 기동정지계획(Unit Commitment Planning)은 시간대 별로 각 발전기 운전 상태를 결정하며, 두 번째인 경제급전계획(Economic Dispatch Planning)은 가동되고 있는 발전기가 얼마만큼의 출력을 가지고 발전하는 것이 효율적인 발전량인가를 결정하는 문제이다. 위에서 전자는 정성적인 제약을 포함하며 후자는 계량적인 제약을 갖는다. 자세한 문제에 대한 수학적인 모델은 참고문헌을 참조하길 바란다[4].

4. 상황변화를 고려한 제약만족문제의 표현

4.1. 전력생산계획을 위한 제약만족문제

전력생산계획 문제를 논리제약프로그래밍으로 표현하게 되면 두 가지의 모델링이 필요한데, 즉 건력생산 가능 발전기 구성을 위한 정성적 모델링과 경제적 전력생산을 위한 계량적 모델링으로 구성한다. 논리제약 프로그래밍으로의 구현은 두 모델링의 인터페이스로서 COMMIT라는 데이터베이스 자원확인자(Assertion)을 사용하며, 일단 발전에 참여할 발전기 후보가 결정되면 데이터베이스에 저장되고 이것을 계량적 모델링에서 해법에 이용된다.

- 경제적 전력생산 문제(Economic Dispatch Problem)

경제적 전력생산문제는 수리계획모델로 표현하고 경제적 전력생산 문제는 economic_dispatch로 아래와 같이 표현된다.

```
economic_dispatch - objective(OBJ OPER, SETUP, P1, P2, ..., Pn),
demand_con(demand1, dReq1, P1, P2, ..., Pn),

reserve_con(reserve1, rReq1, P1, P2, ..., Pn),

minimum_lower_con(P1, P2, ..., Pn)
```

두 가지의 항을 갖는 목적함수식은 술어(Predicate) objective에 의해서 생성된다. 전력수요 제약은 demand_con에 의해서 만들어지며, 순동 예비력 제약과 최대, 최소 출력 및 램핑율 제약은 reserve_con에 의해서 표현된다. 그리고 각 발전기들은 최소량 이상을 발전해야 하는 제약이 있어야 하는데, 그것은 minimum_power_con 술어에 의해서 생성된다

● 발전가능 발전기 구성문제(Unit Commitment Problem)

발전가능 발전기 구성문제는 정상적인 제약을 갖고며 주로 탐색(Search) 방법으로 시스템을 처리한다. 이 문제를 논리프로그래밍으로 표현하면 commitment로 아래와 같이 표현된다.

```
commitment - including_unit(U1), including_unit(U2), ...,
including_unit(Un),
including_unit(U) - unit(U, U_CHAR, U_MARG, U_MAXP, U_MINP,
U_RAMP, ...),
find(UList, DEM_REQ, RES_REQ, U),
update(U, DEM, RES)
find(List, REQ1, REQ2, U) - member(U, List), disj_operate(U, List,
REQ1, REQ2),
```

각 발전기들은 속성(Attribute)을 가지고 있는데 술어 unit는 데이터베이스로부터 최대 발전량, 최소발전량, 램핑율, 발전량에 대한 한계비용 함수식 등을 가져온다. 그리고 find는 발전기마다 속성을 고려해서 발전가능 발전기 집합을 구성하는데, 이러한 절차는 전력수요와 예비력 등을 고려하면서 이루어진다.

● 발전기 기계적 안정성 제약(Unit Mechanical Safety Constraint)

발전가능 발전기 구성문제는 발전기 기계적 안정성 제약인 정상적인 제약을 갖고 발전가능 발전기 집합을 제조정하는 절차를 수행한다

```
unit_select - forall(prefer(U1, U2, T), unit_select(U1, U2, T))
unit_select(U1, U2, T) -
assign(T, U1, U1_CHAR, U1_MARG, U1_MAXP,
U1_MINP, U1_RAMP),
assign(T, U2, U2_CHAR, U2_MARG, U2_MAXP,
U2_MINP, U2_RAMP)
efficiency_select(T, U1_MARG, U2_MARG),
efficiency_select(T, MARG1, MARG2) - not(MARG1==MARG2), 1
efficiency_select(T, MARG1, MARG2) - efficient(MARG1, MARG2)
unit_safety - forall(unit(U), safety_check(U)),
safety_check(U) - assign(U, STATList, ...),
down_check(U, STATList, DN_SAF_VAL),
value_evaluate(U, DN_SAF_VAL),
up_check(U, STATList, UP_SAF_VAL),
value_evaluate(U, UP_SAF_VAL)
```

unit_select는 초기단계에서는 발전기 두 개의 속성을 비교해서 효율적인 발전기를 선정한다. unit_safety는 전체 계획기간동안에 발전시스템의 상태를 추적하면서 기계적인 안전성, 즉 발전기의 기동 및 정지상태별로 “좋은(GOOD)”, “보통(FAIR)”, “나쁨(BAD)” 등을 점검하여 늘 안전성을 극대화하는 방향으로 이끌어 나간다.

4.2. 규칙을 이용한 상황변화의 표현

전력생산을 위한 발전시스템에서는 발전기가 갑자기 정지하거나 한 여류에 전력수요의 갑작스런 급등으로 여유 발전기를 선정해서 발전에 참여시킨다든지, 혹은 전력공급이 충분하거나 발전하려고 예상했던 발전기를 발전가동을 정지시키는 일들이 발생하게 되는 변화가 종종 있다. 이러한 경우에는 빠른 시간 내에 발전가능한 발전기를 선정하고 발전하고 있는 발전기들의 발전량을 제조정하는 일이 시급한 일이다. 반대로 정지할 발전기를 선정하는 일도 어느 발전기를 선정할 것인가, 그리고 이것이 다른 발전기의 발전량에 얼마만큼 변화를 줄 것인가를 결정하는 일도 중요한 문제일 것이다 이러한 상황변화를 시스템에 입력하는 방법은 사용자가 쉬게 다룰 수 있고 표현하는데 어려움이 없어야 한다.

앞에서 설명한 변화를 가지고 시스템이 수행하는 규칙을 표현한 예는 아래와 같다.

```
change_out(unit) if
it_is_included_in(DEL-List)
it_is_included_in(List) if
positive(SUP, DEM) and
have(RES) and
positive(status_up_unit, good)
change_on(unit) if
it_is_included_in(ADD-List)
it_is_included_in(List) if
```

```

positive(DEM, SLP) and
have(RES) and
positive(status_down_unit, good)
exchange(unit1, unit2) if
it_is_included_in(EXCH-List)
it_is_included_in(List) if
positive(DEM, SLP) and
have(RES) and
positive(status_up_unit1, good)
positive(status_down_unit2, good)
    
```

5. 실험적 결과분석

5장에서는 계량적인 제약 하에서 경제적 전력생산을 하는 목적식과 발전기의 기계적 안정성을 고려하면서 발전가능 발전기 집합을 구성하는 정성적 문제의 목적식을 갖는 전력생산계획 문제를 전력회사의 실험자료를 가지고 적용함으로써 이 연구에서 제안한 문제구조의 우수성을 보이고자 한다.

입력자료를 설명하면, 전력생산계획문제의 전통적인 문제에서는 764개의 연속변수 dp_{it} 와 제약식도 896개 이므로, 미진행성을 선형성으로 단순화한다고 하여도 분지 및 한계(Branch and Bound) 방법으로 푸는 경우에는 해답을 구하는 시간이 200시간에서 250시간(PC Pentium 300Mhz 기준) 정도로 매우 힘들고 복잡한 문제이지만, 이 연구에서 제한한 구조에 위하면 변수는 86개이며 제약식도 규칙으로 표현된 제약을 포함하여 46개에 지나지 않는다. 발전용량이 100MW 이상인 발전기 중에서 표본으로 15대의 화력 발전기를 가지고 24 시간대에 적용하였다. 입력자료는 발전기별 기계적 특성과 초기운전 상태를 나타내는 자료로 최대발전량(Pow_max), 최소발전량(Pow_min), 위급시 발전량(Pow_emer), 램핑율(Ram_max), 최소가동시간(Min_up), 그리고 최소정지시간(Min_down) 들이다. 그 외에도 발전기별 비용 계수 자료와 전력수요와 순동 예비력 등이 있는데 이에 대한 자세한 내용은 참고문헌에 잘 나와 있다[4].

두 가지 목적을 갖는 전력생산계획 문제를 문제 표현은 UNK-OPT로 표현하였으며 제약만족 프로그래밍 언어인 ILOG Solver 4.3에 의해서 프로그래밍 되었다. ILOG Solver는 제약만족 문제를 프로그래밍 시에 유용한 함수와 클래스를 포함한 C++ 라이브러리이다. 앞장에서 설명한 경제적 생산계획 문제, 발전가능한 발전기 구성문제, 발전기 기계적 안정성 제약을 가지고 앞의 입력자료를 통해서 <표 2>과 같은 결과를 얻었다.

아래 결과에서 보듯이 발전기의 기동 및 정지가 시간에 따라 변동이 거의 없음을 알 수 있다. 이는 경제적 효과에다가 기계적인 안정성이 고려되었음을 알 수 있다. 그리고 발전기 마다 발전량을 최대로 하지 않는 이유는 짧은 시간 내에 발전량을 변화시킬 수 있는 총량이 바로 순동 예비력이기 때문에 이의 제약을 받기 때문이다. 그리고 모든 발전기는 짧은 시간 내에 변화시킬 수 있는 발전량은 최대발전량에 한정되어 있음을 알 수 있다.

<표 2> 발전기별 경제적 전력생산량 계획결과

		(단위 MW)														
발전기 일련 번호	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
2	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
3	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
4	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
5	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
6	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
7	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
8	50	120	120	0	0	0	0	60	0	130	0	0	250	250	200	
9	50	120	120	0	0	0	0	60	0	130	0	0	250	250	200	
10	50	120	120	0	0	0	0	60	0	130	0	0	0	350	300	
11	100	200	120	0	0	130	0	60	0	130	0	0	250	250	200	
12	50	200	120	0	0	130	0	60	0	130	0	0	250	250	200	
13	50	200	120	0	0	130	0	60	0	130	0	0	250	250	200	
14	50	200	120	0	0	130	0	60	0	130	0	0	250	250	200	
15	50	120	120	170	170	130	0	60	0	130	0	0	250	250	200	
16	50	120	120	170	170	130	0	60	0	130	0	0	250	250	200	
17	50	120	120	170	170	130	0	60	0	130	0	0	250	250	200	
18	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
19	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
20	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
21	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
22	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
23	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	
24	50	120	120	0	0	130	0	60	0	130	0	0	250	250	200	

그리고 발전시스템에서 아래와 같은 상황변화를 입력해 보았다

- 발전기 14호기가 기계적 이상으로 오후 1시부터 정지계획
- 하절기 15시부터 17시까지 전력이 모두 40MW 씩 증가할 것으로 예측됨

위의 상황변화를 감안한 입력자료는 아래와 같다.

DEL-List = {14}, t >= 13.

DEM = {15:40 for t=15, 1560 for t=16, 1560 for t=17}.

4장에서 설명한 규칙이 추론되어 DEL-List를 가지고 14호기 발전기를 13시 이후부터 “기동상태”를 “정지상태”로 전환하며, ADD-List를 규칙에 의해서 생성한다. 규칙은 unit_select를 통해서 발전가능 발전기 집합 11호기, 12호기를 선정하는데 그 이유는 대상이 되는 발전기 중에서 250MW를 공급하기 위해서 대상이 되는 발전기가 7, 9, 11, 12 호기이며, 이 중에서 11, 12 호기가 생산비용 비용식이 가장 낮음을 알 수 있다. 그래서 11, 12호기를 ADD-List에 넣는다. 그리고 15시부터 17시까지 전력의 필요분 40MW는 전력예비력을 감안하여 새로 발전기를 가동하지 않고 위급시 발전량을 고려해서 발전기를 선정한다. 아울러, DEM 전력 수요 변화자료를 가지고 economic_dispatch에서 demand_con의 속성값을 대치시킨다. 이러한 과정을 통해서 상황변화를 감안한 전력생산량을 다시 계획한 결과가 <표 3>이다.

<표 3> 상황변화를 감안한 전력생산량 계획결과

(단위: MW)

발전기 시간	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	50	200	130	0	0	130	0	60	0	130	130	120	250	250	200
13	50	200	120	0	0	130	0	60	0	130	130	120	250	250	200
14	50	200	120	0	0	130	0	60	0	130	130	120	250	0	200
15	50	120	120	130	130	130	0	60	0	130	130	120	250	0	200
16	50	120	120	130	130	130	0	60	0	130	130	120	250	0	200
17	50	120	120	130	130	130	0	60	0	130	130	120	250	0	200
18	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
19	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
20	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
21	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
22	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
23	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200
24	50	120	120	0	0	130	0	60	0	130	130	120	250	0	200

6. 결 론

본 연구는 다중목적에 갖는 규칙기반 제약만족문제의 지식표현에 중점을 두었으며 그 문제구조를 갖는 전력공급계획 문제에 적용하여 많은 실험결과를 통해서 본 연구에서 제시한 지식표현이 탄력적인 전력공급 계획 수립에 매우 효과적임을 입증하였다.

본 논문의 연구결과를 정리하면 다음과 같다.

- 정성적인 제약식을 기존의 계량적인 모델에 접목시킬 수 있는 지식표현 제시
- 상황적인 변화를 감안하여 스케줄링을 다시 세울

수 있는 아키텍처 구축

- 다중 목적을 갖는 문제구조를 위한 제약만족 해결기법 개발

본 논문에서 제시한 문제구조는 산업현장에서 자주 등장하는 구조이며 여기서 개발된 규칙기반 제약만족 해결기법은 상황적인 변화에 능동적으로 대응하며 해법을 제공하는 전자상거래의 해법에 매우 큰 응용력이 있다고 사료된다

참 고 문 헌

- [1] Geoffrion, A. M., The SML Language for Structured Modeling, *Operations Research*, 40(1), 1992.
- [2] Jo, G., M. Yun and J. Yun, Constraint Logic Programming : An Integration of AI with OR, *Korean Intelligent Systems*, Vol.2, No.2, 1993
- [3] Kendrick, D. A., A Graphical Interface for Production and Transportation System Modeling : PTS, *Computer Science in Economics and Management*, 4, 1991
- [4] Kim, Chulsoo, Knowledge-based Approach for Solving Short-term Power Scheduling in Extended Power Systems, *Journal of the Korean OR & MS* 23(2), 1998.
- [5] Lee, H., R. M. Lee and G. Yu, Constraint Logic Programming for Qualitative and Quantitative Constraint Satisfaction Problems, *Decision Support Systems* 16, 1996.
- [6] 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(5), 1995



김철수

e-mail cskim@inl.a.ac.kr

1986년 고려대학교 통계학과 졸업(학사)

1988년 KAIST 경영과학 졸업(공학석사)

1996년 KAIST 경영정보공학 졸업(공학박사)

1996년~1999년 원광대학교 경영학과 조교수

1999년~현재 인하대학교 경영학부 조교수

관심분야 : 인터넷 지식벤처, 무선인터넷 데이터 콘텐츠 설계