

▣ 응용논문

ILOG를 이용한 항공기 운항 일정계획 시스템 개발에 관한 연구  
 - A Study on Development of the Aircraft Scheduling System Using ILOG -

윤 중 준\*

Yun, Joung Jun

이 장 춘\*\*

Lee, Jang Chun

이 화 기\*\*\*

Lee, Hwa Ki

**Abstract**

In this paper, the aircraft route scheduling consists of two steps; the pairing generation and the aircraft assignment. The pairing generation is heuristic approach how flights construct pairings from and to the main base. The aircraft resource assignment which is to establish a aircraft route schedule is constructed by constraint satisfaction problem technique.

ILOG solver and schedule is used to solve this problem. In addition, a aircraft route reschedule process against departure-arrival delay or aircraft maintenance schedule change is added by partial constraints satisfaction problem technique. Also, Gantt chart made by ILOG views is used to show aircraft route schedule results more easily.

**1. 서 론**

일반적인 항공기 운항 일정계획 문제는 항공사가 운항하려는 비행편(flight leg, A도시에서 B도시까지 이동하는 하나의 단위)을 효과적으로 항공기에 할당하고자 하는 것이다. 이를 위해 배당할 좌석 수를 고려하여 주 취항지에서 할당할 항공기 기종을 선택한 후, 선택한 항공기 기종에서 항공기를 비행 편에 할당한다.

본 연구에서는 항공기 기종이 확정된 상태에서 출발공항(연구에서는 서울)을 주기점으로 하는 Pairing(주어진 기점을 출발하여 각종 제약조건을 만족하면서 여러 도시를 거쳐 다시 최초 기점으로 돌아 올 때, 그 일련의 비행편의 순서)을 구성한 후 각 비행편을 항공사의 가용 항공기 자원에 효과적으로 할당할 수 있도록 하는 방법을 적용하여 항공기 운항 경로를 생성하도록 하였다. 연구에서는 항공사의 운항도시, 운항시간, 운항횟수가 결정된 초안 일정계획안을 기초데이터로 하였다. 기초 자료가 되는 초안 일정계획안은 노선 및 비행 횟수 등의 계획안으로서 과거의 영업 일정계획 자료를 보완, 계절별 수요변화, 노선의 연결, 항공기 자원 성능, 승무원 이력, 취항도시의 수요 증감, 목적지 선호성, 정치사회경제 상황을 반영한 주력시장 동향, 경쟁사의 동향을 고려하여 작성된다.

특히 기초 자료로서의 비행편은 일정계획 운영 부서의 항공기 운항 계획(Aircraft Rotation)

\* 인하대학교 산업공학과 박사과정

\*\* 대우정보 시스템 기술연구소

\*\*\* 인하대학교 산업공학과

등 기술적 측면에서 운항정비 부분과 검토 내용을 근거로 마케팅부서와 협조하여 결정되며, City-Pair Block Time(항공기의 운항시간으로, 출발 후 목적지에 도착하여 승객을 내릴 수 있는 장소까지의 이동시간)을 고려할 경우, 정비 운항 승무원 측면에서의 해당 부서별 검토의견, 지상조업(G/H), 객실 서비스의 문제점 유무 확인, 각 취항지의 Slot Time 획득 가능성(항공기가 공항에 출발 및 도착할 수 있도록 허가 받은 시간대 획득 가능성), 운영시간대의 과밀지역(중복)여부와 수요가능성, 운항규정(항공사 자체, 행정당국), 항공기 기종별 기술적 데이터(특정 운항 경로 상의 최대 운항 거리 제한, 항속 거리), 국내선과 국제선 자격변경 소요시간(최저 1시간), 공항별 제약요인 저축여부 확인을 고려하여 수립하게 된다. 이러한 초안 일정계획안의 데이터를 토대로 Pairing을 작성하여 최소의 항공기 대수로 운항할 수 있게 항공기를 할당하는 시스템으로서 항공기 운항 일정계획이 요구된다.

또한, 실제 일정 운영 중에는 정비 보전업무나, 날씨 등의 이유로 선행 일정 지연으로 인한 연쇄 지연, 특정수요의 연결 운송 등의 경우로 인해 통보된 타임테이블(Timetable)상의 일정 중 특정편의 시간대를 조정하는 경우가 발생된다. 이를 위해서 특정편 또는 해당편의 착발 시간대를 확인하고 변경시의 근접 시간대에 속하는 타 비행편의 일정을 확인하고 항공기 운항 일정계획 상황을 수정하게 된다. 따라서 항공기 운영 중 예기치 못한 상황에 발생하는 비행편의 연기 및 취소가 되었을 경우를 고려하고 반영하는 재일정계획 부분이 추가되어야 한다.

현재 국내의 항공사에 이용되는 시스템은 항공기 자원에 대한 운항 일정계획이 항공사의 이익과 밀접한 관계가 있음에도 불구하고 문제의 접근이 어려워 기존에 수립된 일정계획 내용을 토대로 일정계획 전문가에 의한 수 작업을 통하여 갱신되어 왔다. 따라서 항공기 운항 재 일정계획을 위해서 계획 수립과 변경에 시간과 인력을 불필요하게 많이 소요하게 되었으며, 돌발적인 상황에 대처하는데 시간과 노력이 필요 이상으로 소요되었다.

본 논문은 수 작업으로 행해지는 부분을 최소화하고, 항공기를 가장 효율적인 운항 경로에 배치하여 항공사의 불필요한 자원 낭비를 제거하는 해법을 제약 만족 기법과 발견적 기법을 사용하여 제시하고, 시스템에 GUI(Graphic User Interface)를 적용하여 지속적인 일정계획 관리를 할 수 있도록 하며, 적은 인원으로 돌발적인 상황에 효율적으로 즉각 대처할 수 있도록 함으로써 항공사의 경쟁력 향상을 도모하려 한다.

본 논문의 구성은 1장에서 서론을, 2장에서는 항공기 자원 할당을 위한 최근연구와 본 시스템을 위해서 사용된 ILOG 라이브러리에 관한 내용과 ILOG가 기반으로 하는 CSP 이론을 정리하였다. 3장은 초안 일정계획 데이터를 기반으로 하여 Pairing을 구성하는 알고리즘과 Pairing 활동(Activity)을 항공기 자원(Resource)에 할당하는 CSP모델 링과 해의 탐색과정을 소개하였고, 일정계획 알고리즘을 적용한 실행예제를 보여준다. 4장에서는 비행편의 변경으로 인한 재일정계획 알고리즘에 대해서, 마지막으로 5장에서는 결론을 제시하였다.

## 2. 이론적 고찰

### 2.1 항공기 운항 일정계획

1978년 이전까지 항공사가 운항할 수 있는 경로는 고정된 항로만을 운항해야만 하였으나, 미국의 규제완화조치 이후 항공기 스케줄링이 항공사의 이익과 밀접한 관계를 갖게 되었으며 이 분야에 대한 연구가 많이 이루어져 왔다. 항공기 경로 문제(Aircraft Routing Problem)는 주어진 경로에 항공기를 배정하는 문제로 문제를 단순화 시켜 주로 Set Partitioning Problem으로 모델링이 되어 분지 한계법 등의 최적화기법을 이용하여 해를 구하였으나, 항공사의 규모가 커지고 다양한 제약조건이 추가됨으로서 문제의 크기가 커짐으로 인해 만족할 만한 시간 내에

효율적인 해를 구할 수 있는 최적화기반-발견적 해법에 연구의 중점이 이루어지고 있다. 발견적 해법으로는 Column Generation Method와 일정 범위를 초과하는 경로를 제거하면서 경로를 구성하는 Column Screening Method등이 있다.[3]

Abara[1]는 이익최대화, 비용최소화, 항공기이용률 최대화의 다양한 목적함수를 포함한 항공기배정문제를 0-1정수계획법으로 다루었고, Subramanian [13]은 같은 문제에 대해서 출발시간 시간창(Time Window)이 없는 다품목 네트워크 흐름 문제를 제시하고 발견적 분지 한계법을 사용하였다.

Daskin and Panayotopoulos[6]는 Lagrangian Relation을 이용하여 효율이 높은 상한 값을 구한 후 분지 한계법 등의 방법을 통해 가능 해를 구하는 발견적 해법을 제시하였으며, 시간대(Time Period)를 나누어 문제의 크기를 줄이는 방법을 시도하였다.

## 2.2 제약 만족 기반 기법

제약 만족 기반 기법(CSP : Constraint Satisfaction Problem) 개념은 인공지능 분야에서 연구의 대상이 되어 왔다. CSP의 실용적인 중요성은 O.R.에서 특정한 일정계획 문제, 타임테이블링 문제, 그리고 다른 여러 조합최적화(Combinatorial Problem, NP-Hard Problem)문제에서 발생하는 많은 문제들이 CSP 모델링으로 간단하게 표현될 수 있기 때문이다.

CSP는 유한 이산 도메인( $D = \{d_1, d_2, d_3, \dots, d_n\}$ )을 가지는 변수( $V = \{v_1, v_2, v_3, \dots, V_n\}$ )들의 집합과 이들 변수들 사이의 제약 조건( $C = \{c_1, c_2, c_3, \dots, c_n\}$ )들의 집합으로 문제를 정형화한 후 일치성 검사(Consistency Checking)와 경험적 검색 알고리즘(Heuristic Search Algorithm)등을 적용하여 문제를 해결하는 방법이다.[10][12]

CSP는 변수  $V_n$ 에 대해 각 변수에 들어갈 값이 도메인  $D_n$ 라고 하면 지수의 비율인  $O(V_n!)^{D_n}$ 의 계산량을 갖기 때문에 변수나 도메인이 큰 경우에는 많은 시간이 소요된다. 이러한 문제에서 각 변수를 전향검사(Forward Checking)하여 제약조건의 일치성(Consistency)을 검사하고 이에 따라 도메인 여과(Domain Reduction)기법을 적용하여 백트래킹(Backtracking)을 할 경우에는 전체 탐색 공간의 크기를 줄일 수 있다.[11]

제약 만족 기법에서 가장 효과적인 방법 중에 하나는 도메인 여과 기법이다. 도메인 여과 기법은 제약 조건에 의해 변수가 갖는 도메인의 불필요한 부분의 여과, 즉 그 제약 조건과 관련되어 있는 변수의 도메인 축소를 유도한다. 따라서 도메인 여과 기법을 이용하는 제약 만족 문제에 있어서 탐색할 공간(Search Space, Solution Space)이 축소되어 문제 해결을 위한 시간을 효과적으로 줄일 수 있게 된다. 그러므로 해를 구할 경우에 불필요한 백트래킹을 하지 않게 되므로 효과적인 해의 탐색시간(실행시간)을 기대할 수 있다.[15]

불필요한 도메인을 제거하는 도메인 여과 기법은 제약조건의 일치성 검사로 이루어진다. 일치성 검사는 제약조건의 불일치성(Inconsistency)을 일으키는 부분을 찾아 불일치를 유발하는 도메인을 제거함으로써 탐색공간, 도메인의 일치성을 유지하도록 한다. 이러한 일치성 검사를 위해서 일반적으로 제약조건 네트워크를 구성하게 된다. 제약조건 네트워크는 유향성 네트워크로 표현이 되며 제약조건이 적용된 노드(Node)에서 가지(Arc)를 거쳐 제약조건을 다른 노드에 전파하여 충돌된 제약조건의 도메인을 제거함으로써 제약조건의 일치성을 유지한다. 이러한 제약 조건 전파로 인한 네트워크 상에서 제약조건의 일치성은 Arc Consistency 검사로 이루어진다. [11]

일반적으로 CSP를 해결하는 기법은 가능해 집합을 나열하여 가능해나 최적해를 찾아내는 것이다. 문제들을 수학적인 프로그래밍 또는 모델링을 통한 방법이 아닌 CSP 모델로 모델링하고 해를 찾는 이유로는 첫째, 문제 표현이 좀더 문제 그 자체에 가깝게 나타나기 때문이다. 즉 CSP의 변수들은 문제에서의 개체(entity)에 직접 대응되며 제약 조건이 수학적 부등호로

나타내어질 필요 없이 논리적인 표현으로 표현된다는 점이다. 둘째, 이러한 형식적인 표현 면에서 간단하고 이해하기 쉬울 뿐만 아니라 좋은 경험적 탐색 방법의 선택도 직관적으로 이루어질 수도 있다.[12]

### 2.3 ILOG

최근 제약조건 프로그래밍 도구의 개발과 함께 문제를 CSP로 표현하고 해결하는데 더욱 용이하게 되었다. 첫 번째 상업 제약조건 프로그래밍 도구는 Prolog를 확장한 CHIP(Constraint Handling In Prolog)을 기반으로 한 것이다. 이처럼 Prolog와 같은 논리 언어에 제약조건을 다루는 기능을 확장한 제약 논리 언어(Constraint Logic Programming)는 논리 프로그래밍 분야의 중요한 연구 분야이며, C++ 라이브러리인 ILOG Solver가 또 다른 유용한 제약조건 프로그래밍 도구이다.[5]

ILOG Solver의 주요한 특징은 제약조건 만족 프로그래밍(Constraint Programming)을 객체 지향적 환경으로 통합한 것이다. 결과적으로 Solver의 중요한 관점은 단지 Scalar변수가 아닌 객체를 이용하여 제약조건 만족 프로그래밍을 직접 표현 가능하게 한 것이다. 또한 Prolog를 기반으로 하는 다른 프로그래밍 언어와는 다르게 백트래킹을 선택적으로 사용할 수 있기 때문에 non-deterministic algorithm과 deterministic algorithm을 쉽게 통합할 수 있으며, 빠른 속도로 다양한 형태의 일정계획 알고리즘을 생성하고, 여러 가지로 변경하여 실험하는 것을 가능하게 한다.

ILOG Scheduler는 Activity의 공유, Activity에 의해 공유되는 자원측면에서 일정계획 제약조건 표현을 가능하게 하는 C++ 개발 라이브러리이며 ILOG Solver에 추가되어 사용하는 제약조건문제 해결을 위한 C++ 라이브러리이다.

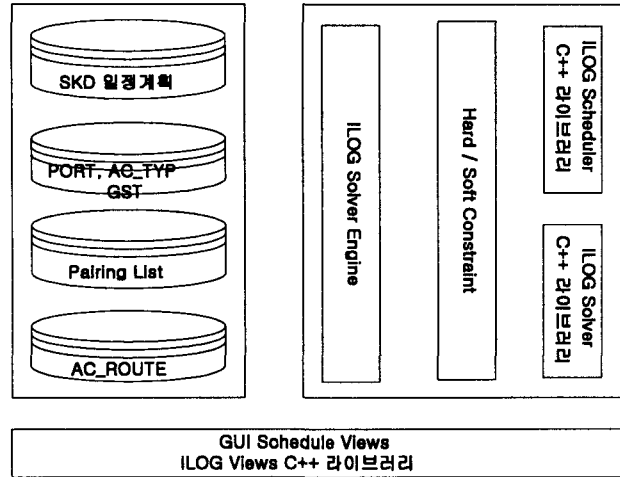
다시 말해서 ILOG Scheduler는 문제의 해를 구하는 S/W이다. 자원과 자원을 사용하고 공유하는 Activity로서 일정계획 문제를 표현할 수 있도록 라이브러리를 사용자에게 제공하며, 클래스를 이용하여 제약조건 변수를 구현하고, 이런 제약조건 변수 사이에 일정계획 제약조건을 정의한다.

## 3. 항공기 운항 일정계획

각 계절·분기별로 이용승객의 수와 추이가 차이를 보이기 때문에 각 항공사들은 초안 일정계획을 기간별(계절별)로 작성하게 된다. 본 논문에서는 K항공사의 실제 초안 일정계획 데이터를 가지고 비행편들을 항공기에 할당하기 위해서 Pairing을 구축하고 Pairing 목록을 이용하여 항공기 운항 일정계획 시스템을 구축한다.

그림 3.1과 같이 항공기 일정계획을 수립하기 위해서 사용되는 기초 데이터로는 항공기 기종 정의(AC\_TYP), 공항 정의(DEP\_SKD\_C, ARR\_SKD\_C), 공항별 CURFEW(소음규제 등에 의한 공항의 이용제한), 구간/기종별 Block Time, 공항별 GST/GT(Ground Standard Time/Ground Time, 비행편이 연결될 때 소요되는 최소 연결시간)가 제공되고, Pairing 구축을 위한 초안 일정계획 데이터베이스로 구성된다.

항공기 할당에 필요한 자료로는 항공기의 고유 ID, 항공기 운항 순서, 해당 운항에 포함되는 비행편의 ID로 구성된다. 초안 일정계획 데이터를 통하여 서울을 주기점으로 하는 Pairing 결과는 Pairing 목록 데이터베이스로 저장되고, Pairing 목록을 항공기에 할당한 운항 경로는 임시 AC\_ROUTE 데이터베이스에 저장된다.



<3.1> 항공기 운항 일정계획 시스템도

### 3.1 Pairing 생성 단계

Pairing을 생성하기 위해 고려되는 비행편은 항공사의 일주일간 자료(계절별로 주간 일정은 같은 패턴을 이룬다)이며, Pairing 생성은 최소연결시간(GST/GT)을 가진 두개 이상의 비행편들로, 최초 비행편의 출발공항(서울)과 최후 비행편의 도착공항(서울)이 같은 공항이 되도록 하여 Pairing 목록을 작성한다.

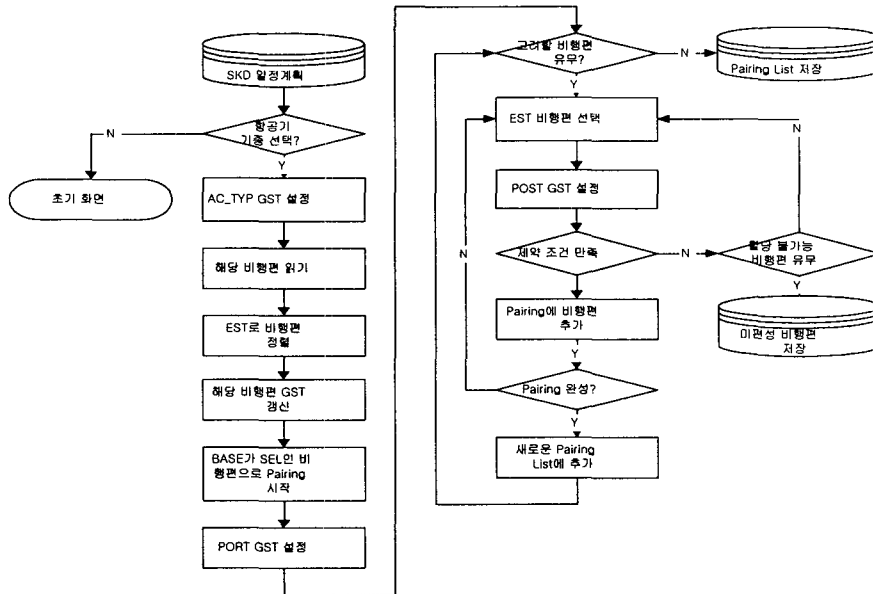
3.1.1절의 Pairing 생성 알고리즘을 이용하여 초안 일정계획에서 얻은 비행편의 자료를 가지고 서울을 주기점으로 하는 Pairing을 구성하게 된다. Pairing을 구성하는 비행편간의 최소연결시간(GST/GT)이 최소화될 수 있도록 Pairing 목록을 작성하고, 얻어진 결과를 간트차트 형태로 그래픽(GUI) 환경 하에 출력하게 된다.

#### 3.1.1 Pairing 생성 알고리즘

Pairing 생성 알고리즘 단계는 다음과 같다.

- 단계1 : 항공기 기종 선택 및 연결 항공기 최소연결시간 설정
- 단계2 : 출발시간이 가장 빠른 순서로 비행편 정렬
- 단계3 : 출발공항이 서울인 가장 빠른 비행편 선택
- 단계4 : 연결 비행편의 도착공항 최소연결시간과 항공기 최소연결시간을 비교하여 가장 긴 연결시간으로 전체 최소연결시간 재설정
- 단계5 : 이전 비행편의 도착공항과 일치하는 연결 비행편의 출발공항과 같은 가장 빠른 비행편을 선택
- 단계6 : 도착공항이 서울이면 Pairing 목록으로 저장하고 단계 3, 4를 반복한다. 모든 Pairing이 생성되면 종료한다.

그림 3.2는 서울을 시작지점으로 하여 공항 및 항공기 최소연결시간 제약조건을 만족하면서 다시 서울 시작지점으로 돌아오는 Pairing을 생성하는 과정에 대한 알고리즘 흐름도를 보여준다. 이때 Pairing 생성에 실패한 비행편에 대해서는 미편성 비행편 데이터베이스에 저장되도록 한다. Pairing 생성단계에서 Pairing생성에 이용되지 못한 비행편들은 항공기 자원 할당 단계에서 밀집도가 가장 작은 항공기 자원을 선택하여 시간과 공간적 제약조건을 만족하는 비행편과 연결되도록 한다.



<그림 3.2> Pairing 생성 흐름도

### 3.2 항공기 자원 할당 단계

Pairing활동의 표현은 최초 비행편의 출발공항이 서울이고 연결 비행편들과 최종 비행편의 도착공항이 서울을 이루게 되므로 각 Pairing간의 공간적 제약조건을 제거할 수가 있고, 항공기 자원에 할당하는 Pairing 활동이 단일 작업으로 표현이 되므로 출발시간과 Pairing지속시간으로 표현하여 Pairing 활동들에 대해 모델링을 한다. 따라서 단일 Pairing의 처음 비행편의 출발시간과 Pairing의 마지막 비행편의 도착시간을 활동의 출발시간, 지속시간으로 하여 시간제약식을 만족하는 항공기 대수에 필요 수만큼 운항 경로를 생성하게 된다.

#### 3.2.1 시간 및 자원 제약조건 모델링

다음 제약조건 모델링은 ILOG Scheduler 라이브러리를 이용하여 Pairing 활동의 출발시간과 Pairing 지속시간 제약조건과 항공기 자원 제약조건을 CSP모델로 모델링 한 것이다.

- 시간 제약조건 모델링
 

```
IlcIntervalActivity Activity(IlcSchedule, Activity_Duration)
Activity.setStartTime(Activity_StartTime)
```
- 항공기 자원 제약조건 모델링
 

```
IlcUnaryResource Resource(IlcSchedule))
m.add(Resource.makeTimetableConstraint())
```

#### 3.2.2 변수 및 값을 선택하는 순서 및 탐색 순서 알고리즘

항공기 자원 할당을 위한 값의 부여 방법은 값을 바인딩 할 변수의 선택 순서(Variable Ordering)와 하나의 변수 내 도메인 중에서 바인딩 할 값을 선택하는 순서(Value Ordering)에 따라 결정된다. 따라서 할당 우선 순위는 FIFO(First In First Out)에 의해 도착시간이 빠른 것이 먼저 수행하도록 하여 변수의 선택시 도착시간이 빠른 활동 순으로 이루어지게 된다.

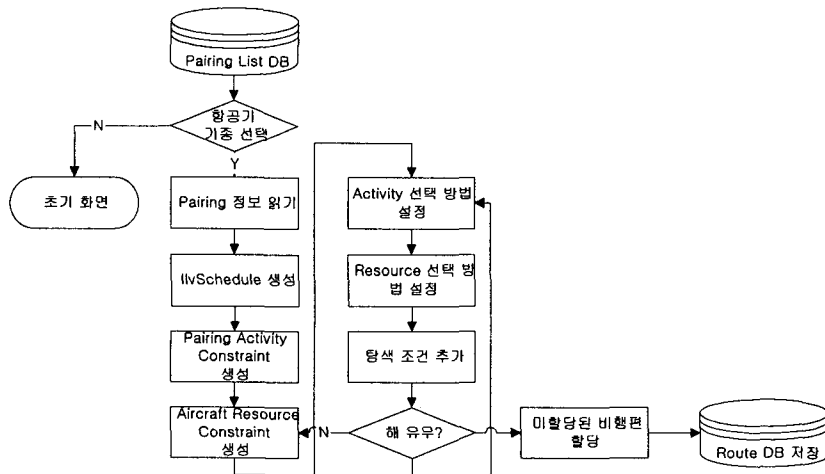
Pairing 활동과 항공기 자원 할당을 위해 값을 할당할 변수의 선택 순서와 변수가 가지는 도메인 중에서 할당할 값을 선택하는 순서 알고리즘 단계는 다음과 같다.

- 단계1: Pairing 활동의 출발시간과 Pairing 지속시간 제약조건 추가
- 단계2: 항공기 가용 자원 타임테이블 제약조건 추가
- 단계3: 선택된 항공기 자원을 이용하는 순서가 결정되지 않은 활동들 중 제일 먼저 수행할 작업을 선택
  - 절차1: EST(Earliest Start Time) 활동 중 EET(Earliest End Time)을 가진 활동
  - 절차2: EST(Earliest Start Time) 활동 중 LST(Latest Start Time)을 가진 활동
  - 절차3: LET(Latest End Time) 활동 중 EST(Earliest Start Time)을 가진 활동
  - 절차4: LET(Latest End Time) 활동 중 LST(Latest Start Time)을 가진 활동
- 단계4: 순서가 결정되지 않은 활동들이 이용하는 항공기 자원들 중 가장 결정적인 자원 검사 방법 선택
- 단계5: 모든 활동의 할당 순서가 결정되면 종료

알고리즘 단계 3의 절차 1~4는 순서가 결정되지 않은 활동들 중에서 가장 먼저 수행할 활동을 선택하는 방법을 4가지로 분류해 놓았다. 본 연구에서는 절차 1의 방법에 의한 활동의 순서가 결정되지 않을 경우, 해당 작업 또는 활동에 대해서는 다음 절차에 의한 방법을 통하여 활동의 순서를 결정하게 된다.

단계 4의 항공기 자원 중에서 “결정적 자원(Critical Resource)”의 선택적 방법은 MinSlack[8] 알고리즘을 이용한다. 결정적 자원은 일정기간동안 하나의 자원이 다른 자원들보다 더 많이 사용되는 자원을 말한다. 즉, 일정기간동안 자원의 효율성과 자원에 대한 수요를 비교하여 최소의 여유를 갖는 자원을 선택하게 된다. 여유에 대한 정의는 일정기간동안 작업의 자원 요구량(이 기간동안 수행되어야 할 작업의 수행시간의 합)과 가용 자원량의 차이로 정의된다.

그림 3.3은 항공기 자원의 할당 흐름도이다. 제약 만족 기법을 적용한 시스템의 효율을 위해서 Pairing 목록을 최초 출발공항, 마지막 도착공항, 연결 비행편으로 분류하여 항공기에 할당하고 각각의 부분 경로를 구축해 나가는 과정을 통하여 모든 Pairing과 Pairing에 고려되지 않았던 비행편들이 경로를 이루어나간다. 마지막으로 Pairing을 이루지 못한 비행편에 대해서는 모든 Pairing 활동이 할당 된 후, 순차적 검색을 통하여 시·공간적인 제약조건(출발공항과 출발시간)을 만족하는 비행편을 밀집도가 가장 작은 항공기 자원에 할당하는 방법을 사용하였다.



<그림 3.3> 항공기 자원 할당 흐름도

### 3.3 실행 예제

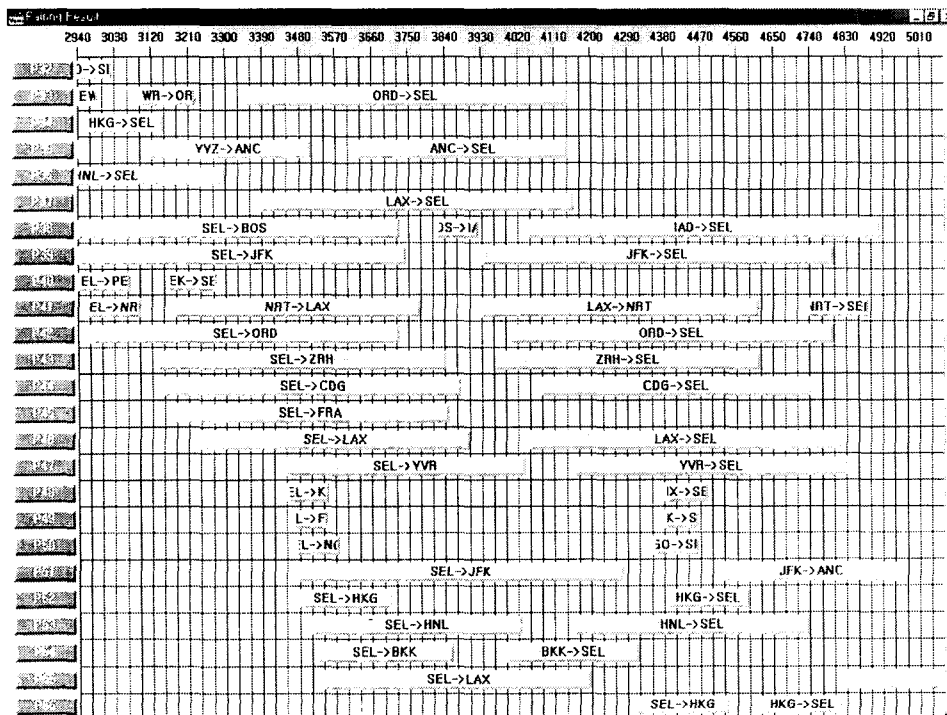
모든 일정계획을 수립하기 위한 정보의 입출력은 데이터베이스 ODBC Driver를 통하여 이루어지며, 사용자 관리기능은 ILOG Views3.0 라이브러리를 이용하여 일정을 항공기 일정계획 기간동안 관리하도록 하였다. 또한 시스템의 주화면 및 간트차트 기능 구현을 위해서 ILOG Views 및 Studio 라이브러리를 사용하였다.

본 시스템 적용을 위해서 K 항공사의 2000년 06월 04일부터 06월 10일까지 일주일간의 자료를 가지고 Pairing을 작성하였다. Pairing 생성과 노선 생성을 위해서 선택한 기종은 744기종이고, 항공사의 주 취항지로는 북미와 유럽이며 일주일간 300회 이상 운항을 하는 것으로 나타났다.

#### 3.3.1 Pairing 생성 결과

Pairing을 생성하기 위해서 우선 항공기종을 선택하여 데이터베이스에서 쿼리(SQL Query)을 통해서 원하는 정보를 추출하고, 이를 토대로 Pairing 생성 알고리즘을 이용하여 Pairing을 생성하며 이때 사용되는 비행편의 출발 및 도착 시간은 서울 현지 시각을 기준으로 한다.

그림 3.4는 Pairing 생성결과 화면으로 출력되는 모든 간트바의 출발/도착 시간은 서울 시각을 기준으로 나타낸다. 각 간트바는 하나의 비행편을 나타내고, 출발도시와 도착도시를 간트바에 표시하여 사용자가 전체적으로 출발·도착 도시와 시각을 한눈에 알아볼 수 있도록 하였다. 간트차트의 P0, P1, P2, ..., Pn은 일주일간 비행편을 Pairing으로 이룬 Pairing 목록을 보여주는 것이다. Pairing 목록은 두편 이상의 비행편으로 이루어져 있는데, 각각의 비행편들은 출발도시와 도착도시, 출발시간과 도착시간 등을 기입하여 사용자가 한눈에 직접 알아 볼 수 있도록 했으며, 간트바의 위 부분에 확대하고자 하는 부분을 마우스(Mouse)로 지정함으로써 확대하여 Pairing을 관리하도록 하여 사용자 편의성을 도모하였다.



<그림 3.4> Pairing 생성 결과



### 3.3.2 항공기 운항 일정계획 결과

그림 3.5는 744기종에 대해서 생성된 Pairing자료를 가지고 항공기 자원에 할당하여 항공기 경로를 생성한 화면을 보여준다. 2000년 06월 04일부터 06월 10일까지 모든 초안 일정 계획을 바탕으로 Pairing을 생성해서 만든 항공기 경로를 순차적으로 수평 스트롤 바를 이용하여 모든 일정계획 관리를 한눈에 관리하도록 하였으며, 화면상의 각 간트바는 하나의 비행편으로 아래에 출발도시와 도착도시를 표시하도록 하였다. R0, R1, R2, R3, ..., Rn은 K사 744항공기가 일주일간 담당해야 할 운항 경로를 나타내며, 간트바는 해당 운항 경로에 Pairing을 이룬 비행편들을 나타내고, 비행편 하단에 출발도시와 도착도시를 표시하도록 하였다. 스프레드시트의 흰 부분과 음영부분은 주야간을 나타낸다.

Route	6월 5일	6월 6일	6월 7일	6월 8일	6월 9일	6월 10일
R0	SEL	SEL	HKG		HKG	SEL
R1		SEL	SEL		SEL	SEL
R2	K071	SEL			SEL	SEL
R3		SEL	SEL		SEL	SEL
R4	K076	SEL			SEL	SEL
R5	NRT	NRT	SEL		SEL	SEL
R6	SEL	SEL			SEL	SEL
R7			JFK	JFK		SEL
R8	NRT		LAX	LAX	NRT	NRT
R9	SEL	SEL			SEL	SEL
R10	SEL		CN	CAI	DXB	DXB
R11		SEL	SEL		SEL	SEL
R12	K062	ORF			SEL	SEL
R13	HKG	SEL			LAX	LAX
R14						

<그림 3.5> 항공기 운항 일정계획 결과

## 4. 항공기 운항 재일정계획

일정계획에서 초기해의 변경으로 인한 연쇄효과(Ripple Effect)를 미리 예측할 수 없기 때문에 재일정계획(Reactive Scheduling)의 어려움이 발생하게 된다. 연쇄효과란 초기해의 변화로 인하여 발생하는 일정 계획된 다른 비행편들의 변화를 말하며, 일정계획 요소의 변화로 인하여 초기해가 변화하는 상태가 되면, 그에 따른 비행편들의 제약조건과 비행편에 할당되는 항공기 자원이 불규칙하게 발생하기 때문이다. 일반적인 재일정계획은 일정계획 요소의 변화로 발생하는 초기 일정계획의 결과에 대한 불일치성을 제거하여 초기 일정계획의 결과를 유지 보수하는 것이다.

시스템의 속도와 복잡도 측면의 효율성을 얻기 위해서 재일정계획 문제에서도 이전의 일정 계획과 같이 탐색공간을 줄이기 위해 CSP해결 기법을 적용하여 정형화하였다. 항공기 자원에

할당할 비행편(Activity)을 변수로 지정하고, 이 변수에 대한 도메인을 필요한 자원(Resource)에 할당한다.

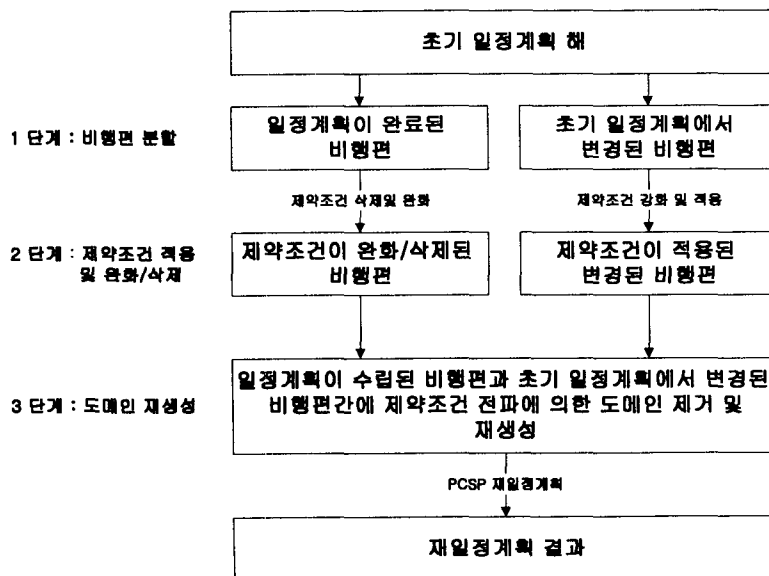
재일정계획에서 고려할 대상은 변경된 비행편과 재일정계획 범위 안에 포함되는 비행편을 항공기의 제약조건으로 정하고, 비행편이 변경될 경우 초기 해의 변경은 자원을 할당한 비행편에서 발생하게 된다. 따라서 이러한 일정계획의 변화는 해당 비행편의 삭제, 삽입, 변경을 통하여 재일정계획을 수립할 수 있다.

해당 비행편이 삭제될 경우는 일정계획 초기 해에서 일정계획 된 비행편이 삭제되는 것으로 비행편에 할당된 자원은 기존에 삭제된 도메인 영역으로 반환된다. 비행편의 삽입은 기존의 비행편에 새로운 비행편을 첨가하는 것인데, 비행편 삭제의 경우와 달리 여러 가지 상황이 발생할 수 있고 재일정계획의 경우 주요 문제점으로 나타난다. 마지막으로 초기 일정계획 해에서 비행편 변경은 삭제와 삽입을 일련의 순서로 수행하는 것으로 표현할 수 있으며, 이 경우 변경해야 할 노선의 비행편에 대한 제약조건과 관련된 일정계획 요소들을 삭제하고 수정하여 일정계획의 불일치성을 해결한다.

변경된 비행편으로 인한 재일정계획을 수립할 경우, 해당 활동(비행편)들이 출발, 도착예정시간을 갖기 때문에 재일정계획의 범위는 변경된 비행편의 변경시간 이후이어야 한다. 그리고 재일정계획의 대상은 일정계획 변경 시간 이후의 비행편과 항공기에 한정하는 규칙이 적용되어야 한다.

일정계획 모델링의 객체화 표현 측면에서 재일정계획의 요소를 CSP로 모델링하여 변수와 도메인, 그리고 제약조건으로 표현할 경우에는 비행편(활동)과 항공기(자원)에 관한 제약조건을 적절하게 표현하여 적용하기 어렵다. 변수의 경우에는 재일정계획의 모든 비행편들과 항공기를 적용할 경우 도메인의 증가상태를 유발할 수 있고, 또한 많은 제약조건을 표현할 경우 표현의 복잡성이 증가하게 된다.

그림 4.1은 운항 재일정계획 단계를 도식화한 것이다. 일정계획 요소의 변화는 재일정계획을 활성화시키고 최소 범위 제약 만족 재일정계획 알고리즘을 적용할 때까지 3단계를 거쳐서 일정 계획된 초기 비행편들과 일정계획의 요소가 변경된 비행편에 대한 도메인을 제거하여 탐색공간을 줄인다.



<그림 4.1> 운항 재일정계획 메커니즘

재일정계획의 첫 번째 단계는 비행편의 분할 단계부분에 해당되는데, 이 단계에서는 초기 일정계획의 해로부터 일정계획 된 비행편과 변경된 비행편으로 분할 관리한다. 두 번째 단계에서 초기의 제약조건 적용과 완화 단계에서 이미 제약조건이 적용되어 있는 일정계획 된 비행편과 변경된 비행편 간의 충돌로 인한 해를 발견하지 못할 가능성을 낮추기 위해서 제약조건 완화 및 삭제기법을 적용하여 해의 발견 가능성을 높인다.[4]

제약조건 완화 및 삭제 기법은 초기 일정계획을 수립하기 위해서 사용된 제약조건들을 삭제하고 일정계획을 재수립하기 위한 항공기 자원 선택방법과 비행편 또는 Pairing 목록을 선택하기 위한 방법들에 관한 제약조건들을 삽입하는 방법을 사용하였다. 마지막 단계에서 제약조건이 변경되어 일정계획 된 비행편과 초기 일정계획의 비행편 및 자원에 대한 제약조건을 고려하여 제약조건 전파(Constraint Propagation)에 의한 유효한 도메인을 구한다.

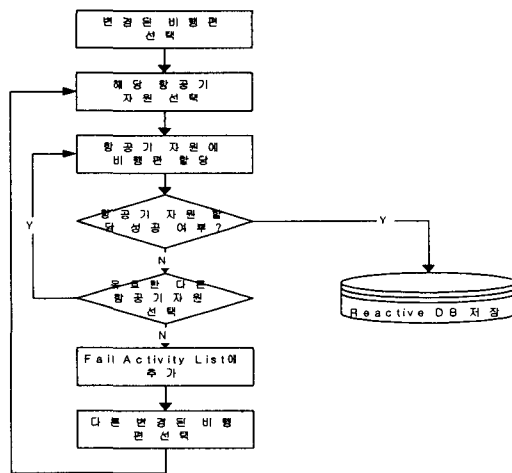
4.1 운항 재일정계획 알고리즘

재일정계획을 위한 시스템에서는 초기 일정계획을 유지하는 측면에서 초기해의 최소 변화, 부분해, 속도 등의 융통성을 요구하게 된다. 그러나 시스템이 이러한 융통성들간의 복잡성 때문에 균형을 이루면서 재일정계획을 유지하는 것은 매우 어렵다. 재일정계획에 대한 접근 방법을 제약 조건 만족 해결 기법을 통하여, 최소 범위 재일정계획 모듈을 모델링하는 방법으로 시행하였다.

재일정계획을 위한 시스템 구축에서 초기해의 변화가 일정계획 결과에 혼란을 줄 수 있으므로 초기해의 변경을 최소화하는 방법으로 접근하여 수행한다. 초기 일정계획 해의 정보를 이용하여 초기해의 변화를 최소화하고 재일정계획 도메인 제거 단계의 결과를 가지고 초기 일정계획 결과의 정보에 따라 변경된 비행편에 할당된 항공기 자원에 대한 우선 자원 할당의 순위를 두도록 하여 초기 일정계획 해의 변화를 최소화하도록 하였다.

CSP에서 복잡한 제약조건, 또는 도메인의 부족으로 인하여 제약조건을 만족하는 변수가 존재하지 않는 경우가 발생할 경우, 규모가 큰 문제에서 발생한다면 이에 따른 적당한 조치나 혹은 적당하다고 할 수 있는 부분해(Partial Solution)를 제공해야 한다. 이와 같이 부분해를 얻는 문제를 CSP에서는 부분 제약 만족 문제(PCSP : Partial Constraint Satisfaction Problems)라고 한다. 이러한 PCSP의 해결은 시스템의 모델링을 통한 문제 해결에 접근해야 한다.[2][14]

그림 4.2와 같은 부분 제약 만족 문제 기법을 사용하여 재일정계획에서 점진적인 제약조건 만족을 적용하여 부분 해를 구하였는데 이는 초기 해를 근간으로 하여 변경된 비행편 들을 적용하여 해를 얻게 된다.



<그림 4.2> 부분 제약 만족 기법을 이용한 재일정계획 알고리즘

### 4.2 재일정계획 예

본 연구에서는 항공기 운항 일정계획에서 생성된 초기해를 가지고 실험한다. 항공기 운항 재일정계획 범위는 변경된 비행편에 의거하여 해당 비행편을 담당하는 항공기 자원과 변경된 비행편 이후 비행시간대로 정한다. 재일정계획에 고려할 운항경로를 R4, R7, R9로 설정하고 변경된 비행편의 설정을 변경시간대가 6월 5일 정오(12:00)를 포함하는 비행편에 대해서 90분 비행시간을 추가하였다. 따라서 T90은 항공기의 기술적인 문제로 인한 지연시간이 90분이 추가된다는 의미를 나타내고 있다. 그림4.3은 그림3.5를 재일정계획 한 결과 화면이다.

Route	6월 5일	6월 6일
R3	K069	K122 K129
R4	K076 PEK SEL KIX K054 T-90 HNL HNL K055 SEL SEL K100	
R5	K047 NRT NRT SEL K113 BOS BOS	K114
R6	K052 DFW DFW ATL ATL DFW DFW K053 K096 K097	
R7	K058 T-90 JFK JFK K059 K104	
R8	K045 NRT LAX K046 NRT NRT SEL SEL K091	
R9	K050 T-90 LAX LAX K051 K121	
R10	K085 GAI GAI DXB DXB K087 K119	BKK
R11	COT SEL SEL FUK K063 FUK SEL SEL K131	FRA
R12	K082 ORF ORF K063 SEL SEL K108	JFK
R13	K066 HKG SEL SEL K048 LAX LAX K084 K117	K093
R14	K083 LHR LHR K056 K106 SEL SEL HKG	K10
R15	K011 SEL SEL ANC ANC YVZ YVZ K102 ANC ANC	
R16	K064 GRU GRU K065 C12 LAX	
R17		

<그림 4.3> 항공기 운항 재일정계획 결과

재일정계획 수립을 위한 탐색의 순서로는 변경된 비행편이 가장 빠른 순서와 비행시간이 가장 짧은 순서, 밀집도가 가장 작은 순서로 탐색을 하였다. 따라서 변경된 비행편으로 인하여 후속 비행편의 출발시간이 최소한의 변경이 되거나 변경된 비행편의 영향으로 최소한 변경 비행편이 발생하도록 하였으며, 변경 제약 비행편으로 인한 변경 영향으로부터 빨리 벗어나도록 하였다.

재일정계획에서 항공기 자원을 소비하는 비행편의 시간이 겹치는 정도인 밀집도가 작은 순서에 의해 탐색을 할 경우에 할당 실패확률을 줄일 수 있고, 기간이 작은 순서의 탐색을 수행함으로써 일정계획의 속도를 높이도록 하여 백트래킹 횟수를 줄이도록 하였다. 따라서 성공할 비행편을 우선 탐색하여 할당하도록 하여 재일정계획의 효율성을 높이도록 하였다.

## 5. 결 론

다양한 산업영역에서 수행되는 일정계획 문제를 해결하기 위하여 AI분야에서는 최근 CSP(Constraints Satisfaction Problem)개념이 소개되었고, 그 효율성이 입증되고 있다. 제약조건을 기반으로 한 방법은 문제를 CSP로 정형화한 후 제약조건에 따른 일치성 검사 및 경험적 탐색 방법(Heuristic Searching Technique)을 적용하여 문제의 해를 효율적으로 구하게 된다.

CSP 라이브러리(ILOG Solver, Scheduler)를 사용하여 객체 지향적인 기법을 사용함으로써 제약식·지식 표현이 간결해지고 제약조건의 적용 부분과 객체화를 통하여 시스템의 유지보수 및 문제의 표현 요소들을 파악하기 용이한 방법으로 표현하였다.

본 연구에서는 항공사의 운항도시, 운항횟수, 그리고 운항시간이 결정된 상태의 초안 일정계획안의 기초데이터로부터 실제 항공기에 할당하기 위한 Pairing을 구성하고, 작성된 Pairing으로부터 ILOG를 이용하여 효율적인 방법으로 항공기 운항 경로를 수립할 수 있게 하였다.

항공기 운항 재일정계획을 위한 시스템 구축에서는 초기 해의 변화를 일정계획 결과의 사용에 혼란을 줄일 수 있는 초기 해의 변경을 최소화하는 부분 제약 만족 기법(PCSP)으로 접근하여 수행하였다. 즉, 초기 일정계획 해의 정보를 이용하여, 초기 해의 변화를 최소화하도록 하는 재일정계획 알고리즘에서는 재일정계획 단계에서 수행한 도메인 제거 결과를 가지고 일정계획된 비행편에 대하여, 초기 일정계획 결과의 정보에 따라 변경된 비행편에 할당된 항공기 자원에 대해서만 자원 할당의 우선 순위를 두도록 하여 초기 일정계획 해의 변화를 최소화하도록 하였다.

추후 연구로는 실시간 일정계획, 항공기 보전 일정계획, 승무원 일정계획과 연계된 일정계획의 시스템 통합 구축이다.

## 참고문헌

- [1] Abara, J., "Applying Integer Programming to Fleet Assignment Problem", *Interfaces*, vol. 21, no.4, pp. 20-78, 1989.
- [2] Arie M.C.A. Koster, Stan P.M. van Hoesel, Antoon W.j.Kolen., "The partial constraint satisfaction problem : Facets and lifting theorems", *Operations Research Letters*, vol. 23, pp.89-97, 1998.
- [3] Anbil, R., "Recent Advances in Crew-Pairing Optimization at American Airlines", *Interfaces*, vol. 21, no. 1, pp. 62-47, 1991.
- [4] Beck J. C., and M. S. Fox, "Mediated Conflict Recovery by Constraint Relaxation", *Proceedings of American Association for Artificial Intelligence Workshop on Models of Conflict Management in Cooperative Problem Solving*, Seattle, WA, 1994.
- [5] Claude Le Pape, "Three Mechanisms for Managing Resource Constraints in a Library for Constraint-based Scheduling", *Proceedings of the INRIA/IEEE Conference on Emerging Technologies and Factory Automation*, 1995.
- [6] Daskin, M. S. and N. D. Panayotopoulos, "A lagrangian Relaxation Approach to Assignment Aircraft to Routes in Hub and Spoke Network", *Transportation Science*, vol. 23, no. 2, 1989.
- [7] ILOG, *ILOG Solver manual*, ILOG, 1998.

- [8] ILOG, *ILOG Scheduler manual*, ILOG, 1998.
- [9] ILOG, "Optimization technology white paper", ILOG, 1998.
- [10] Koli Nonobe, Toshihide Ibaraki, "A tabu search approach to the constraint satisfaction problem as a general problem solver", *European Journal of Operational Research*, vol. 106, pp.599-623, 1998.
- [11] Mackworth, A. K., "Consistency in Networks of Relations", *Artificial Intelligence*, no. 8, pp.99-119, 1977.
- [12] Smith, B. M., "A Tutorial on Constraint Programming", *University of Leeds School of Computer Studies Research Report Series*, Report 95, no. 14, 1995.
- [13] Subramanian, R., R. P. Scheff, J. D. Quillinan, D. S. Wiper and R. E. Marsten, "Coldstart : Fleet Assignment at Delta Air Lines", *Interfaces*, vol. 24, no. 1, pp. 104-120, 1994.
- [14] Voundouris C. and E. Tsang, "Partial Constraint Satisfaction Problems and Guided Local Search", *Proceeding of Practical Application of Constraint Technology*, London, pp. 337-356, 1996.
- [15] Wallace, R. J., "Why AC-3 is Almost always Better than AC-4 Establishing Arc Consistency in CSP's", *American Association for Artificial Intelligence*, 1994.