

■ 論 文 ■

COSMOS에서의 Sub Area 운영 알고리즘에 관한 연구 및 평가

Study and Evaluation of Sub Area Linkage Algorithm in COSMOS

이 영 인

(서울대학교 환경대학원 교통관리전공 교수)

김 상 윤

(한국도로공사 도로교통기술원 연구원)

목 차

- I. 서론
 - II. COSMOS의 현황 분석
 - 1. COSMOS의 SA 구성
 - 2. 교차로유형별 기능
 - 3. COSMOS의 SA 관련 알고리즘 현황
 - III. SA 운영 알고리즘의 개선
 - 1. SA 내의 운영 알고리즘
 - 2. SA 결합/분리 알고리즘
 - IV. 실험 및 평가
 - 1. 기능성 평가
 - 2. 효율성 평가
 - V. 결론
- 참고문헌

Key Words : COSMOS, Sub Area(제어그룹), CI(중요교차로), 결합/분리 알고리즘, 대응제어

요 약

COSMOS는 1990년부터 개발되어 현재까지 현장실험 및 시스템 검증과 여러 차례의 기능개선을 통하여 점차 안정화 되어가고 있으며, 그 효율성이 입증되어 서울시 전역으로 확산되고 있는 추세이다. 그러나 현재는 CI의 읍셋패턴이 한쪽 축 방향의 유입/유출 이동류에 의해서만 산출되고 있어, 다른 축방향에 속한 교차로는 자신과 무관한 읍셋패턴에 의해 운영되는 경우가 발생한다. 또한 Sub Area 결합/분리시 CI의 유입/유출 방향의 축에 속한 Sub Area와의 결합/분리만이 가능한 실정이다. 따라서 본 연구에서는 CI의 동서와 남북 축 각각의 유입/유출 방향을 설정하여, 두 축의 읍셋패턴과 현재 운영 주축을 계산하여 설정하도록 하였다. Sub Area 내의 교차로들은 자신이 속한 축의 읍셋패턴에 의해 운영되며, 결합/분리시에는 주축방향간의 '결합'과 이외의 경우의 '연동'의 개념으로 운영되도록 하였다. 개선된 알고리즘은 COSMOS 운영 S/W에 탑재하여, 강남구 10여개 교차로를 대상으로 기능성 평가 및 효율성 평가를 실시하였다. 평가 결과 개선 알고리즘이 기존방식보다 통행시간을 감소시키는 것으로 나타났다.

I. 서론

외국의 경우 Traffic Adaptive Signal System 은 1970년대부터 개발하여 많은 나라에서 운영되고 있다. 그 중에서 가장 널리 운영되고 있는 것은 호주의 SCATS(Sydney Coordinated Traffic System)과 영국의 SCOOT(Split, Cycle, Offset Optimization Technique)이다. 우리나라의 경우 서울시에서 날로 심화되는 교통난 해소를 위해 '91년부터 '93년까지 교통상황에 따른 실시간 신호제어시스템(COSMOS)을 개발하여 운영 중에 있다.

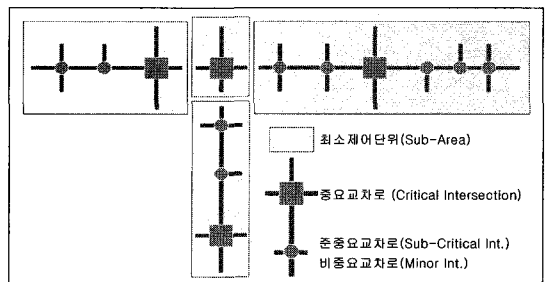
COSMOS(Cycle, Offset, Split Model for Seoul)는 현재까지 현장실험 및 시스템 검증과 여러 차례의 기능개선을 통하여 점차 안정화 되어가고 있으며, 그 효율성이 입증되어 서울시 전역으로 확산되고 있는 추세이다. 그러나 COSMOS에서 현재 운영중인 SA(Sub Area, 제어그룹) 결합/분리 알고리즘은 여러 개의 SA 결합시 연동값이 어긋나게 되어 비효율적인 운영을 하게되는 문제점이 있다. 또한, 유입이동류와 유출이동류의 설정이 한쪽 측만 가능하기 때문에 설정되지 않은 축으로 결합이 사실상 불가능한 실정이다. 이와 더불어 SA내의 교차로 운영도 한쪽 축에 대한 패턴정보를 모두 동일하게 사용하기 때문에 비중요교차로의 경우 자신의 축 방향과는 무관한 패턴정보로 운영하게 되는 등 몇 가지 문제점이 발견되었다.

따라서 본 연구에서는 SA가 1축과 2축의 패턴정보를 모두 생성할 수 있도록 개선하여, SA내 교차로들이 2개의 패턴정보 중 자신의 축 방향에 해당하는 패턴정보를 받아 운영되도록 하였다. 이와 더불어 SA의 결합이 2개 축 모두 가능하도록 하며, 결합/분리시에는 주축방향간의 '결합'과 이외의 경우의 '연동'의 개념으로 운영되도록 하였다. 또한, 여러 개의 SA 결합시 발생하는 연동값 이상 문제를 개선하였다. 개선한 알고리즘을 COSMOS에 탑재하였으며, 수정된 소프트웨어는 실내실험을 통해 디버깅 및 수정·보완을 하였다. 이후 COSMOS 운영 지역인 강남구의 10여개 교차로를 대상으로 현장실험을 수행하였다. 현장실험은 기능성평가와 효율성평가로 나누어 실시하였다.

II. COSMOS의 및 현황 분석

1. COSMOS의 SA 구성

동일 지역내라 하더라도 가로망의 형태나 교통패턴의 변화, 주 교통류 방향 등에 따라 교통류의 연동화에 크게 영향을 미친다. 이러한 점들을 고려하여 몇 개의 유사한 교통패턴을 유지하는 교차로를 묶어 그룹화하고 있으며, 이를 Sub Area(제어그룹)로 정의하고 있다. <그림 1>과 같이 하나의 Sub Area는 반드시 하나의 중요교차로(CI)를 포함하며, 그 이외의 준중요교차로(SCI) 및 비중요교차로(MI)로 이루어진다.



<그림 1> Sub Area 구성

2. 교차로유형별 기능

CI는 각 접근로 검지기에서 DS를 산출하여 실시간으로 주기와 split을 계산한다. 또한 한쪽 축 방향의 유입/유출 이동류가 설정되어 있으며, 유입/유출 이동류의 교통량을 비교하여 그 축 방향의 옹셋패턴을 결정한다. SCI와 MI는 CI의 신호주기와 옹셋 패턴에 의해 운영된다. <표 1>은 교차로 유형별 기능을 요약 정리한 것이다.

<표 1> 교차로유형별 기능

유형	기능
중요 교차로 (CI)	<ul style="list-style-type: none"> ○ 1개 제어그룹 당 1개소 ○ SA 내에 다른 교차로들을 제어 ○ 주기와 split을 계산하고 좌회전 감응제어를 수행하는 교차로
준중요 교차로 (SCI)	<ul style="list-style-type: none"> ○ CI에서 계산된 주기를 사용 ○ CI는 아니나 split 계산 및 좌회전 감응제어를 수행하는 교차로
비중요 교차로 (MI)	<ul style="list-style-type: none"> ○ Sub Area의 나머지 교차로 ○ CI에서 계산된 주기를 사용하며, 사전 계획된 신호시간을 통하여 split 계산 ○ 좌회전 감응제어 혹은 대기행렬 검지를 수행

3. COSMOS의 SA 관련 알고리즘 현황

오프셋값은 <표 2>와 같이 16개의 입력 data가 존재하며, CI에서 계산된 주기와 오프셋패턴에 따라 변경되게 된다. 오프셋의 변경은 유입/유출 교통량 비를 계산하여 연속적으로 3주기 이상 오프셋 변경이 요구될 때 이루어진다. 다음은 COSMOS에서 오프셋을 선택하는 과정을 나타낸 것이다.

- [step 1] 예측된 다음 주기의 포화도에 포화 교통류를 계산하여 유입, 유출 방향의 교통량 산출
- [step 2] 유입/유출 교통량 비(IOR : In/Out Volume Ratio) 계산
- [step 3] 예측된 다음주기의 길이와 유입/유출 교통량 비를 통해 오프셋 패턴 선택
- [step 4] 연속적으로 3주기 이상 오프셋 변경이 요구되면 오프셋 변경을 수행

<표 2> 오프셋값 data 예시

구분	주기1	주기2	주기3	주기4	주기5	주기6
Inbound	10	10	15	20	25	30
Balanced		30	30	30	30	30
Outbound		50	45	40	35	30

오프셋이 선택되면 입력된 오프셋값 data와 메인현시까지의 split 시간을 고려하여 오프셋 전이량을 결정하게 된다. 오프셋의 전이량 만큼 신호현시들을 줄이거나 늘리는 방식으로 오프셋의 전이가 일어난다. 결합오프셋의 적용시에도 이와 같이 전이량의 결정 및 전이과정을 수행하게 된다. 오프셋 전이를 할 경우 <표 3>에서와 같이 주기감소와 증가의 경우에 대한 한주기 동안의 오프셋 전이량 허용치를 설정하고 있다. 이는 급작스런 주기 변화로 인한 교통혼잡을 방지하기 위한 장치이다. 오프셋 전이의 과정은 다음과 같다.

- [step 1] 최소 녹색시간과 주기의 17%를 비교하여 작은 값은 오프셋 전이 사용시간으로 결정
- [step 2] 기존 오프셋 값과 새로운 오프셋 값의 차를 오프셋 전이량으로 결정
- [step 3] 주기를 늘릴 때의 전이 주기수와 주기를 줄일 때의 전이 주기수를 계산
- [step 4] 결정된 주기만큼 늘리거나 줄이면서 현시별 녹색시간을 결정

<표 3> 오프셋전이 허용치

구분	최대 허용치
주기감소	○ 최대 주기의 17%(주기 - 최소녹색시간) 중에서 작은 값
주기증가	○ 최대 주기의 33%

COSMOS에서는 정해진 조건에 따라 SA간의 결합/분리를 결정하며, 결합/분리 수행시의 운영방식은 다음과 같다.

- 결합조건 및 운영방식
 - 조건 1 : 연속적으로 3주기 이상 인접 결합가능 SA와 주기 차가 운영자가 지정한 범위(기정값 ±10초) 이내인 경우
 - 조건 2 : 연속적으로 3주기 이상 인접 결합가능 SA와의 오프셋 패턴(유입, 균등, 유출) 인덱스가 같아지는 경우
 - 앞의 2가지 결합 조건을 검토하며, 결합조건 4가지 옵션으로 결정
 - 0 : 결합하지 않음
 - 1 : 조건 1과 조건 2를 모두 만족하는 경우
 - 2 : 조건 1을 만족하고, 연동방향이 같거나 1레벨 차이일 경우 결합
 - 3 : 연동방향이거나 주기차이에 관계없이 무조건 결합
 - 오프셋값이 제공되는 주방향으로 제어그룹 결합
 - 결합 SA의 CI : 신호주기가 가장 큰 SA를 CI로 설정
 - Offset 값 : 결합 Offset 사용
 - SA 결합 후 지속 주기 수
 - SA 결합한 주기로부터 최소 3주기 동안은 분리조건 1이 만족되더라도 결합 지속하며, 분리조건 2가 만족될 경우는 바로 분리됨
- 분리조건 및 운영방식
 - 조건 1 : 연속적으로 3주기 이상 결합된 SA와의 주기 차가 운영자 지정범위(기정값 ±10초)를 벗어난 경우
 - 조건 2 : 결합된 SA와 오프셋 패턴(유입, 균등, 유출)의 인덱스가 달라지는 경우
 - 앞의 2가지 조건을 검토하여 결합조건 4가지 옵션상황에 따라 결정
 - 0 : 무조건 분리

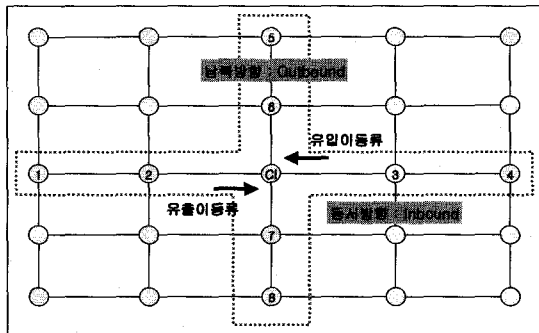
- 1 : 조건 1과 조건 2중에서 하나라도 만족하면 분리
- 2 : 조건 1을 만족하거나 연동방향이 2레벨 차이일 경우 분리
- 3 : 분리하지 않음
- 읍셋 값이 제공되는 주방향으로 SA 분리
- CI : SA의 해당 CI
- 읍셋 값 : 양단 교차로는 결합 읍셋을 사용하지 않음

III. SA 운영 알고리즘의 개선

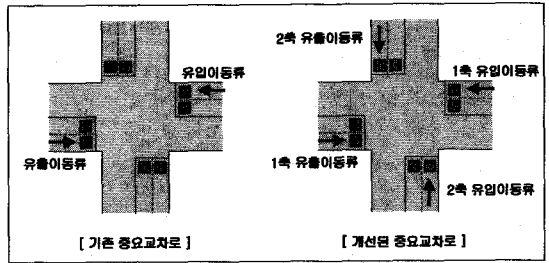
1. SA 내의 운영 알고리즘

현재 알고리즘에서 CI는 한쪽 방향의 읍셋 패턴만을 계산하므로, SA내의 모든 교차로는 그 방향의 읍셋 패턴에 따라 운영되었다. 따라서 SCI 및 MI는 자신의 축 패턴과 무관한 읍셋패턴으로 운영되는 문제점이 발생하였다. <그림 2>에서 CI는 동서방향의 읍셋 패턴만을 계산한다. 만약 동서방향의 읍셋 패턴이 Inbound이고, 남북 방향의 실제 읍셋 패턴이 Outbound일 경우를 살펴보자. 1~4번 교차로는 자신의 읍셋 패턴에 맞게 운영되지만, 5~8번 교차로는 상황이 Outbound임에도 불구하고 Inbound로 운영되게 된다.

개선된 알고리즘에서는 CI의 축을 두개로 설정하여 각각의 축에 대한 읍셋패턴을 계산하고, SCI 및 MI는 두 축중 자신이 속한축의 읍셋패턴 정보로 운영되게 하였다. CI의 개선 내용을 살펴보면, <그림 3>과 같이 한쪽 축의 유입/유출이동류만 설정하던 것을 두 개 축 모두에 대해 설정하도록 수정하였다. 각 축은 1축과 2축으로 정의하였으며, 차로당 평균교통량을



<그림 2> 기존 제어그룹 운영 예시도



<그림 3> 개선 전·후의 중요교차로(CI) 비교

비교하여 실시간으로 주축과 부축을 결정하게 된다. 축결정이 완료되면 주축과 부축별로 읍셋 패턴을 계산한다.

CI에서 주축 및 읍셋패턴 계산은 다음과 같은 단계로 수행된다.

[step 1] 교통량 update 및 평활화

- 1축의 유입교통량과 유출교통량 및 2축의 유입교통량과 유출교통량을 전송받음
- update된 3주기의 1축 유입/유출교통량, 2축 유입/유출교통량을 평활화

[step 2] 요구주기 level 결정

- 기존의 방식과 동일하게 요구주기 level 결정

[step 3] 주 연동축 결정

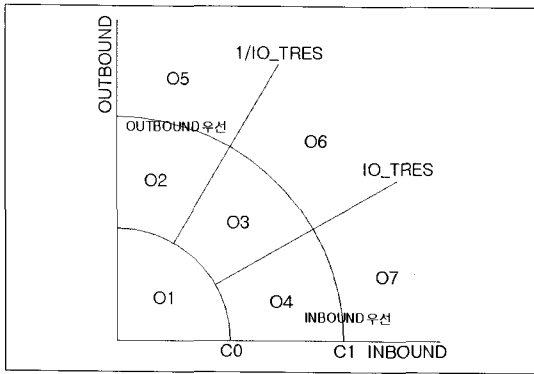
- 1축과 2축의 평균 교통량 계산
 - $vol = (1\text{축유입교통량} + 1\text{축유출교통량}) / 2$
 - $Svol = (2\text{축유입교통량} + 2\text{축유출교통량}) / 2$
 - 현재 교통량 상황에 따른 주축계산(calAxis)
 - $Svol \geq Fvol \times \text{축결정비율}(1.6)$ 일 경우 $\Rightarrow calAxis = 2\text{축}$
 - $Svol < Fvol \times \text{축결정비율}(1.6)$ 일 경우 $\Rightarrow calAxis = 1\text{축}$
- (축결정비율 : 운영자 설정값)

• 운영 주축 계산(Op_Axis)

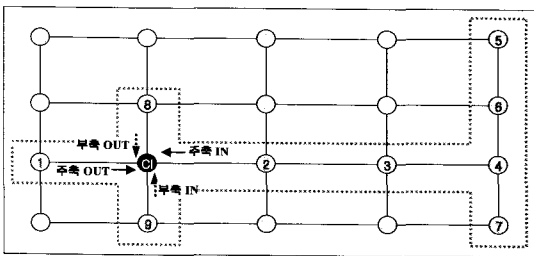
- 연속적으로 3주기 동안 calAxis와 이전주기의 Op_Axis가 다를 경우 운영 주축 변경

[step 5] 읍셋패턴 결정

- 주축과 부축의 유출입비를 각각 계산하고, 이를 <그림 4>와 같이 운영자가 입력한 유출입비율과 비교하여 요구패턴 결정
- 현재 운영되는 읍셋방향과 다른 방향이 연속 3주기 동안 동일하게 요구될 경우에는 읍셋 패턴을 변경



〈그림 4〉 옹셋 선택 개념도

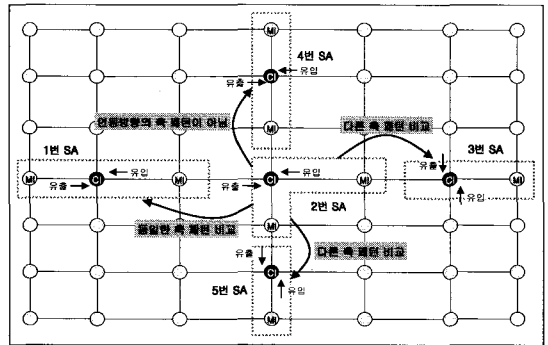


〈그림 5〉 개선된 SA 운영 예시도

CI에서 각 축의 옹셋패턴을 위와 같이 계산하면, SCI 및 MI는 계산된 각 축의 옹셋 패턴중 자신이 속한 축 방향의 옹셋 패턴으로 운영되도록 수정하였다. 〈그림 5〉와 같이 SA가 설정되어 있을 경우, CI의 주축 방향에 위치한 1~4번 교차로는 주축의 옹셋 패턴으로 운영된다. 그러나 CI의 부축 방향에 위치한 8, 9번 교차로의 경우는 부축 방향의 옹셋 패턴으로 운영되도록 하였다. 이때 CI의 주축과 부축 중 어느 축에도 속하지 않는 5~7번 교차로는 주축방향의 옹셋 패턴으로 운영하며, 3→4→6과 7→4→3의 좌회전 교통류를 연동시켜 운영한다.

2. SA 결합/분리 알고리즘

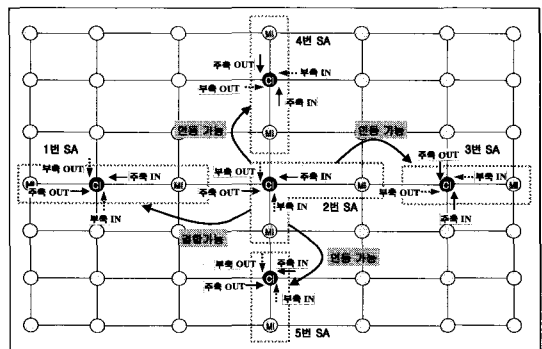
기존의 COSMOS의 S/W는 동서와 남북 축 중 어느 방향으로도 결합/분리를 수행할 수 있다. 그러나 CI의 유입/유출이동류가 한쪽 축 방향으로만 설정되어 있기 때문에 다른 축 방향으로의 결합/분리시 서로 상관없는 옹셋 패턴을 비교하게 된다. 따라서 한쪽 축 방향으로만 결합/분리가 가능하다. 〈그림 6〉에서 2번 SA가 인접 결합 가능한 SA와 결합/분리 조건을 검토할 경우를 예로 살펴보자. 먼저 1번 SA와



〈그림 6〉 현재 결합/분리시 옹셋 패턴 비교 문제점

결합/분리 검토할 경우 SA간의 연동방향은 동서이고, 각 SA에서 계산된 옹셋 패턴도 동서방향의 패턴이므로 아무런 문제가 없다. 그러나 나머지 3, 4, 5번 SA와 결합/분리를 검토할 경우에는 SA간의 연동축 방향과 무관한 방향의 옹셋 패턴을 비교하게 되므로 불합리한 결합이 발생하게 된다.

개선된 결합/분리 알고리즘에서는 CI의 주축과 부축 옹셋패턴을 계산하므로, 이를 이용하여 SA간의 옹셋 패턴 비교시 연동가능한 축방향의 옹셋패턴을 비교하도록 하였다. 또한 결합되는 SA간의 주축과 부축 상황에 따라 '결합' 또는 '연동'의 방식으로 운영되게 하였다. 만약 결합대상 SA가 결합결정 SA의 주축 상에 존재하고, 결합결정 SA 또한 결합대상 SA의 주축 상에 존재할 경우 '결합가능' 상황이 된다. 이때 SA가 결합되면 이를 '결합'이라고 정의하였다. 그러나 결합가능 SA가 결합결정 SA의 주축 상에 존재하지 않거나, 결합결정 SA가 결합가능 SA의 주축 상에 존재하지 않을 경우 '연동가능' 상황이 된다. 이때의 SA 결합을 '연동'이라고 정의 하였다. 〈그림 7〉에서 2번 SA가 인접 SA와 결합/분리 조건을 검토할 경우 1번



〈그림 7〉 수정된 결합/분리 알고리즘 예시도

〈표 4〉 2번 SA와 인접 SA간의 패턴 비교축

인접 SA	위치	주축	패턴비교축	2번 SA의 패턴비교축	비고
1번	2번 SA의 주축방향	동서	주축(동서)	주축(동서)	결합 가능
3번		남북	부축(동서)	주축(동서)	연동 가능
4번	2번 SA의 부축방향	남북	주축(남북)	부축(남북)	연동 가능
5번		동서	부축(남북)	부축(남북)	연동 가능

SA만이 '결합가능' 상황이며, 나머지 SA들은 모두 '연동가능' 상황이 된다.

개선된 CI가 주축과 부축의 읍셋 패턴을 계산하므로, 각 SA간에 결합조건을 검토하는 경우 비교되는 읍셋 패턴은 연동축 방향에 맞는 읍셋 패턴을 비교하여야 한다. 〈표 4〉는 〈그림 7〉에서 2번 SA가 인접 SA와 결합/분리 조건을 검토할 경우, 자신과 인접 SA가 서로 비교하는 축의 읍셋 패턴이다. 2번 SA가 3번 SA와 결합/분리 조건을 검토할 경우, 자신의 주축(동서) 읍셋패턴과 3번 SA의 부축(동서) 읍셋패턴을 비교하여 결정하는 것이다.

인접 SA와 읍셋패턴을 비교하여 '결합가능' 상황에서 결합이 결정되면 '결합'으로 운영되며, '연동가능' 상황에서 결합이 결정되면 '연동'으로 운영된다. '결합'과 '연동'의 운영방식은 약간의 차이가 있다. 먼저 '결합' 상황이 되면 결합을 결정한 SA는 분리전까지 다른 SA와 '결합' 및 '연동' 검토를 수행하지 않는다. 그러나 '연동' 상황에서 연동을 결정한 SA가 3주기 Transition을 마치면 '결합' 상황을 만족하는 다른 SA가 있을 경우에 한해 '연동'을 해제하고 그 SA와 '결합'을 수행하게 된다. 〈그림 7〉에서 2번 SA가 인접 결합가능 SA와 '결합' 및 '연동'을 수행할 때의 운영 방식은 다음과 같다.

- 2번 SA가 1번 SA와 '결합'할 때의 운영
 - 운영 주기 : 1번 SA의 운영 주기
 - 주축 및 부축 읍셋 패턴
 - 주축의 읍셋패턴(동서) : 1번 SA의 주축 읍셋 패턴(동서)
 - 부축의 읍셋 패턴(남북) : 2번 SA 자신의 부축 읍셋 패턴
 - 결합 읍셋값으로 3주기 동안 읍셋 전이 수행

- 분리전까지 다른 SA와 '결합' 및 '연동' 검토를 하지 않음

- 2번 SA가 5번 SA와 '연동'할 때의 운영
 - 운영 주기 : 5번 SA의 운영 주기
 - 주축 및 부축 Offset 패턴
 - 주축의 읍셋 패턴(동서) : 2번 그룹 자신의 주축 읍셋패턴
 - 부축의 읍셋 패턴(남북) : 5번 SA의 부축 읍셋 패턴(남북)
 - 결합 읍셋값으로 3주기 동안 읍셋 전이 수행
 - 3주기 읍셋 전이 완료 후 '결합가능'한 1번 SA와 결합조건 검토
 - 1번 SA와 결합이 가능할 경우 결합을 수행하며, 5번 SA와는 '연동'이 해제됨
 - 1번 SA와 결합 조건이 충족되지 않으면 5번 SA와 분리조건 검토

SA 결합/분리 알고리즘은 다음과 같은 단계로 수행된다.

- ① 결정SA 상태가 SINGLE일 경우
 - 결합하지 않은 상태로써 ④를 수행
- ② 결정SA 상태가 MERGE_AND_HOLD일 경우
 - '결합' 또는 '연동'하여 3주기 전이중인 상태로써 ⑤를 수행
- ③ 결정SA 상태가 MERGE_AND_FREE일 경우
 - 3주기 전이가 완료되면 '결합' 또는 '연동' 중인 대상그룹과 '결합' 및 '연동' 상황을 update
 - 대상SA와의 상황이 '연동'이면, 다른 대상SA와의 결합조건 검토를 위해 ④를 수행
 - 결정SA의 상태가 MERGE_AND_FREE일 경우 ⑥을 수행
- ④ 결정SA의 상태가 SINGLE이거나 '연동'일 경우
 - 주축이 1축일 경우에는 '제어단위 결합/분리 설정 테이블'의 맨 위 대상그룹부터, 주축이 2축일 경우에는 맨 아래의 대상그룹부터 다음의 과정을 수행하여 결합조건을 검토하도록 하였다. 이는 주축 상에 존재하는 대상그룹에게 결합조건 검토의 우선순위를 주기 위한 것이다.

[step 1] 대상SA에 자신이 결합하였거나 대상SA가 자신에게 결합한 경우 다음 SA 검토, 아닐 경우 [step 2]를 수행

[step 2] 결정SA의 상태가 '연동'일 경우 대상SA와의 상황이 '결합' 및 '연동' 가능인지를 검토, 결정SA의 상태가 SINGLE일 경우 [step 3] 수행

- 조건 1 : 결정SA의 주축 상에 대상SA가 존재하는지 check
- 조건 2 : 대상SA의 주축 상에 결정SA 자신이 존재하는지 check
- 조건 1과 2중 하나라도 만족하지 않을 경우 다음 SA 검토, 둘 다 만족하면 [step 3] 수행

[step 3] 대상SA와 주기를 비교하는 함수 수행

- 결합 가능한 주기이면 cycMatch=YES
- 결합 불가능한 주기이면 cycMatch=NO

[step 4] 대상SA와 패턴을 비교하는 함수 수행

- 결합 가능한 패턴이면 patMatch=YES
- 결합 불가능한 패턴이면 patMatch=NO

[step 5] 연속적으로 3주기 동안 cycMatch=YES이고, patMatch=YES 이면

- 결합 읍셋값 계산
- 결합시 신호계획 결정
- 결정SA의 상태를 READY_TO_MERGE로 변경
- 자신이 결합한 대상SA의 번호를 저장
- 테이블의 몇 번째 SA에 결합되었는지 check
- 대상SA에게 자신이 결합한 사실을 통보
- Transition이 완료될 때까지 대상SA의 읍셋값이 변화하지 않도록 함
- '결합' 상황인지 '연동' 상황인지 check

⑤ 결정SA 상태가 MERGE_AND_HOLD일 경우

[step 1] 테이블의 SA중 자신과 결합된 SA check

- target=자신이 결합한 대상SA 번호

[step 2] 결합 후 3주기가 지난 경우

- 결정SA의 상태를 MERGE_AND_FREE로 변경

[step 3] 결합 후 3주기 이내일 경우

- 결합유지 중인 주기 수 update
- 결합읍셋값 재계산
- 결합시 신호계획 재결정

⑥ 결정SA 상태가 MERGE_AND_FREE일 경우

[step 1] 테이블의 SA중 자신과 결합된 SA check

- target=자신이 결합한 대상SA 번호

[step 2] target과 자신의 패턴을 비교

- 결합 가능한 패턴이면 patMatch=YES
- 결합 불가능한 패턴이면 patMatch=NO

[step 3] patMatch=NO 일 경우

- 대상SA에게 자신이 분리된 사실을 통보
- 그룹의 상태를 READY_TO_DIVIDE로 변경

[step 4] 주기비교 함수를 수행하여 target과 자신의 주기를 비교

- 결합 가능한 주기이면 cycMatch=YES
- 결합 불가능한 주기이면 cycMatch=NO

[step 5] 연속적으로 3주기 동안 cycMatch=NO일 경우

- 대상SA에게 자신이 분리된 사실을 통보
- 결정SA의 상태를 READY_TO_DIVIDE로 변경

[step 6] 결정SA의 상태가 MERGE_AND_FREE 상태로 유지될 경우

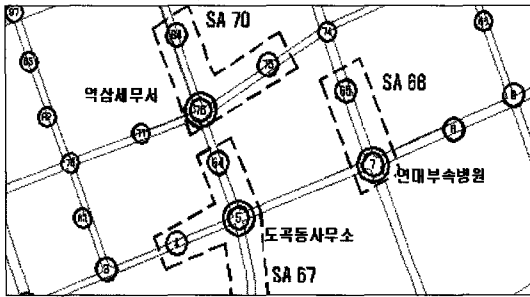
- 결합읍셋값 재계산
- 결합시 신호계획 재결정

IV. 실험 및 평가

본 연구에서는 개선된 SA운영 알고리즘을 현재 운영되는 COSMOS의 S/W 탑재하였으며, 수정된 S/W는 도로교통안전관리공단의 실내 실험 장비를 이용하여 수정·보완을 수행하였다. 현장실험은 강남구 일대의 10여개 교차로를 대상으로 기능성 평가와 효율성평가로 나누어 수행하였다.

1. 기능성 평가

<그림 8>과 같이 도곡동 사무소를 중심으로 한 9개 교차로를 기능성 평가 대상지로 선정하였다. 총 3개



〈그림 8〉 현장 실험 대상지

의 SA를 설정하여 동서측과 남북측으로의 결합/분리를 상황을 분석하였다.

〈그림 9〉, 〈그림 10〉, 〈그림 11〉은 각 SA의 결합/분리 설정 테이블의 입력 DB를 나타낸 것이다. 본 실험은 기능성평가를 주 목적으로 하므로, 결합가능한 주기편차를 20초로 설정하였다. 67번 SA는 68번 및 70번 SA와 결합이 가능하며, 68번 SA와 70번 SA는 각각 67번 SA와 결합이 가능하도록 설정하였다.

〈표 5〉는 각 SA에 속한 CI 및 MI의 측관련 입력 DB를 나타낸 것이다. 이는 CI의 1축 및 2축 설정 정보와 MI의 CI 소속측의 입력 정보이다. MI의 CI 소속측이란, CI의 1·2축 읍셋 패턴중 MI가 따라야할 측을 의미한다. 이외에도 주기레벨과 읍셋값, split data, 주현시 등의 입력DB가 있으나 분량이 많아 생략한다.

68	7	5	-40	10	10	1	3	3	20	0	0
70	76	64	10	10	-50	1	3	3	20	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

〈그림 9〉 67번 SA 제어단위 결합/분리 설정 DB

67	5	7	40	-10	-10	1	3	3	20	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

〈그림 10〉 68번 SA 제어단위 결합/분리 설정 DB

67	64	76	-10	-10	50	1	3	3	20	0	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

〈그림 11〉 70번 SA 제어단위 결합/분리 설정 DB

〈표 5〉 각 SA의 측 관련 입력 DB

67번 SA				
구분	5번(CI)	MI의 CI 소속측		
		4번	6번	64번
1축	유입이동류	동	○	
	유출이동류	서		
2축	유입이동류	남		○
	유출이동류	북		
68번 SA				
구분	7번(CI)	65번(MI)		
1축	유입이동류	동		
	유출이동류	서		
2축	유입이동류	남	○	
	유출이동류	북		
70번 SA				
구분	76번(CI)	75번(MI)	84번(MI)	
1축	유입이동류	동		○
	유출이동류	서		
2축	유입이동류	남	○	
	유출이동류	북		

〈표 6〉 인덱스별 연동방향

주기레벨	1	2	3	4	5	6										
인덱스	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
연동방향	유입	균형	유출	유입	균형	유출	유입	균형	유출	유입	균형	유출	유입	균형	유출	유출

COSMOS에서는 6개의 주기레벨을 가지며, 인덱스는 각 주기레벨에서의 연동방향을 숫자로 나타낸 것이다. 인덱스는 〈표 6〉과 같이 3개의 연동방향을 나타낸다.

기능성평가는 교통상황에 따른 각 측의 읍셋패턴 변화와 결합/분리시 읍셋값이 정확한지를 평가하였다. 먼저 각 SA내 CI의 운영패턴을 살펴보면 다음과 같다.

운영패턴에서 계산 1축은 1축의 요구패턴이며, 운영 1축은 CI가 실제로 운영중인 패턴을 나타낸 것이다. 〈그림 12〉와 〈그림 13〉에서 5번교차로의 제이기 문제로 인하여 계산 1·2축의 인덱스가 3회에 걸쳐 '1' 값으로 떨어지는 현상이 발생하였다. 67번 SA는 초기 70번 SA와 결합하여 12주기까지 유지하여 자신의 계산 인덱스와 다른 70번 SA의 1축 인덱스로 운영되었다. 40~50주기에는 68번 SA와 결합하여

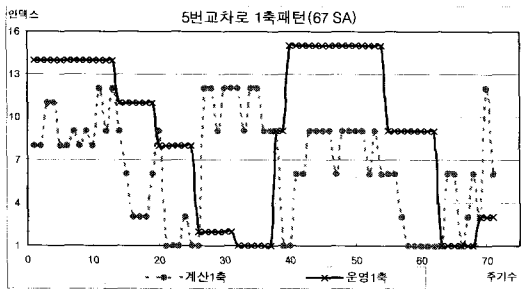
운영되었으며, 이때의 운영 인덱스 또한 자신의 계산인덱스가 아닌 68번 SA의 2축 인덱스로 운영되었다.

〈그림 14〉, 〈그림 15〉의 68번 SA는 10~25주기 경에 TOD로 운영되어 계산축과 운영축이 달라졌으며, 다른 SA와 결합하지 않았다. TOD로 운영된 주기를 제외한 주기동안은 계산 주축에 의해 운영 주축이 합리적으로 결정되어 운영됨을 볼 수 있다.

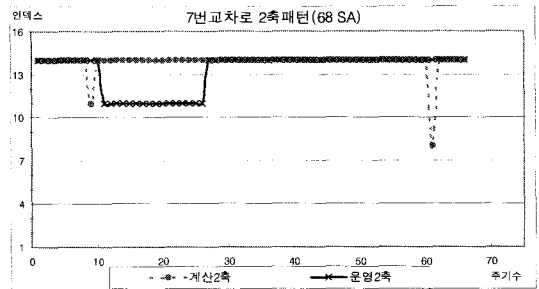
〈그림 16〉, 〈그림 17〉의 70번 SA는 11~17주기 동안 67번 SA와 결합하였으며, 39~46주기에 도 67번 SA와 결합하여 운영되었다.

〈표 7〉은 3개의 SA가 모두 결합하였을 때의 상황을 요약한 것이다. 먼저 68번 SA를 살펴보면, SINGLE인 상황에서 주축이 2축이고(인덱스 14) 부축이 1축

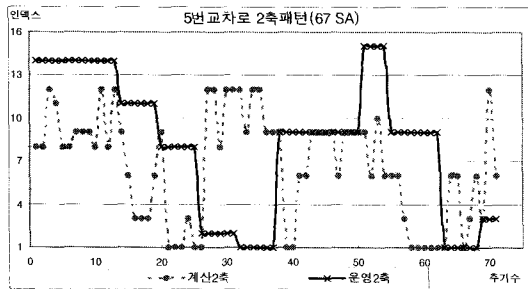
(인덱스 15)으로 운영중이다. 이때 67번 SA가 자신의 주축인 1축 방향으로 68번과 결합하여, 1축의 인덱스는 15가 되었다. 그러나 부축(2축)은 소프트웨어에 미 탑재된 부분 때문에 67번 SA 자신의 부축 인덱스인 9로 운영되고, 운영주기는 동일하였다. 67번 SA가 68번에 결합된 상태에서 70번 SA가 67번에 결합하였으며, 이때 70번 SA의 주축(1축)은 67번의 부축(2축) 인덱스로 운영되었다. 운영 주기는 160초의 동일한 상황이며, 70번 SA의 부축(2축)은 자신의 인덱스인 16으로 운영되었다. 적용읍셋값은 읍셋 입력값 DB와 메인현시의 split 시간을 고려하여 각 교차로가 적용되어야 할 읍셋값을 계산한 값이며, 운영읍셋값은 실험에서 각 교차로에 적용된 실제 읍셋값이다.



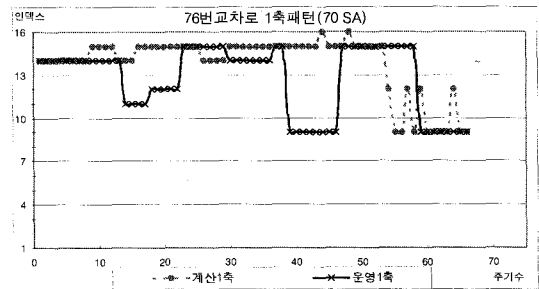
〈그림 12〉 5번 CI의 1축 운영패턴(SA 67)



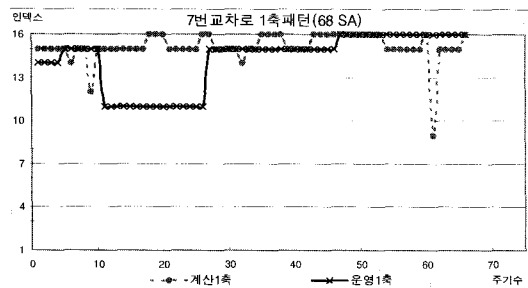
〈그림 15〉 7번 CI의 2축 운영패턴(SA 68)



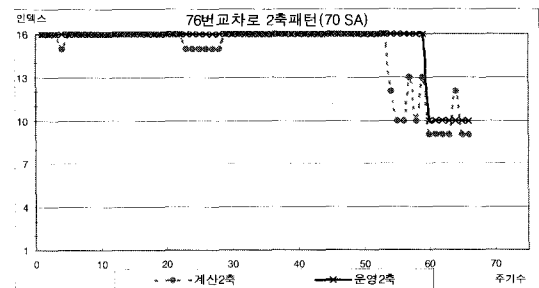
〈그림 13〉 5번 CI의 2축 운영패턴(SA 67)



〈그림 16〉 76번 CI의 1축 운영패턴(SA 70)



〈그림 14〉 7번 CI의 1축 운영패턴(SA 68)



〈그림 17〉 76번 CI의 2축 운영패턴(SA 70)

〈표 7〉 70→67→68번 SA로 결합할 경우 운영상황 (단위:초)

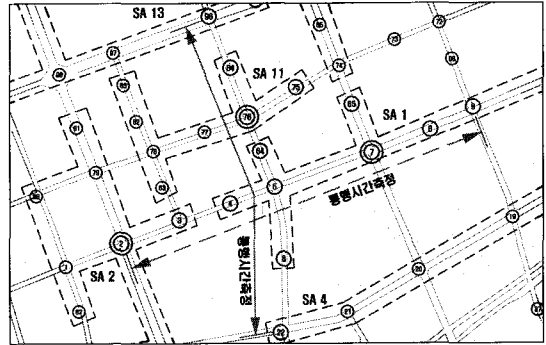
70번 SA(67번 SA와 결합)					
구분	소속축	적용 읍셋값	운영 주기	운영 인덱스	운영 읍셋값
75번	2축(동서)	36	159	16	37
76번	CI	134	159	9	135
84번	1축(남북)	98	159	9	99
67번 SA(68번 SA와 결합)					
구분	소속축	적용 읍셋값	운영 주기	운영 인덱스	운영 읍셋값
4번	1축(동서)	83	160	15	83
5번	CI	60	159	15	61
6번	2축(남북)	144	160	9	144
64번	2축(남북)	4	159	9	5
68번 SA(SINGLE, 주축 : 2축, 1축 인덱스 : 15)					
구분	소속축	적용 읍셋값	운영 주기	운영 인덱스	운영 읍셋값
7번	CI	54	160	14	52
65번	2축(남북)	44	159	14	45

실험결과 SA내 교차로 운영알고리즘의 개선대로 1축과 2축의 교차로들은 자신의 해당축 패턴 인덱스로 운영되었다. 또한 주축과 부축 방향 모두 결합/분리가 가능하였으며, 3개 SA의 연속적인 결합도 가능하였다. 여러개 SA 결합시 읍셋값이 어긋나던 문제점도 해결되었다. 단 주축방향 결합시 부축의 패턴 인덱스 결정이 미진하였으나 기능성평가 후 개선하였다.

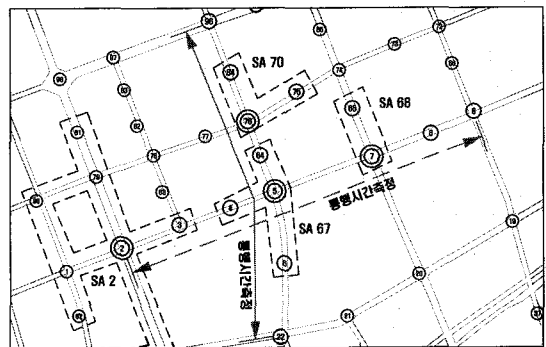
2. 효율성 평가

효율성 평가는 probe 차량을 이용하여, 동서측인 도곡동길 일부구간과 남북측인 논현로 일부구간의 개선 전·후 통행시간을 측정하였다. 이와 함께 center에서의 상황 모니터링 및 자동으로 생성되는 로그파일을 분석하였으며, 분석 MOE는 통과교통량과 대기길이이다. 〈그림 18〉은 기존방식의 실험 네트워크로 현재 경찰청에서 운영중인 형태이며, 1번 SA와 11번 SA는 동서방향으로 설정되어 남북 방향으로는 결합할 수 없다. 〈그림 19〉는 개선방식의 실험네트워크이다.

효율성 평가는 교차로 운영성과 구간 운영성과로 나누어 분석하였다. 교차로 운영성과는 〈표 8〉, 〈표 9〉와 같이 전체 교차로 통과교통량 및 대기길이를 이용



〈그림 18〉 기존방식의 실험네트워크



〈그림 19〉 개선방식의 실험네트워크

〈표 8〉 교차로별 차로당 통과교통량(2시간) (단위:대수)

구분	5번 CI			7번 CI			76번 CI		
	기존	개선	증감 (%)	기존	개선	증감 (%)	기존	개선	증감 (%)
북↓남	531	529	-0.4	561	616	10.7	656	639	-2.6
남↓북	860	876	1.9	1442	1382	-3.8	1043	1162	11.4
동↓서	707	733	3.7	517	581	12.4	400	452	13
서↓동	496	471	-5.0	405	418	3.2	414	431	4.1
계	2594	2609	0.6	2925	2997	2.5	2513	2684	6.8

하여 분석하였다. center의 로그파일은 검지기가 작동중일 경우만 생성되므로 3개의 CI교차로만을 대상으로 하여 분석하였다. 통과교통량 분석은 〈표 8〉과 같이 5번 CI는 0.6%, 7번 CI는 2.5%, 76번 CI는 6.8% 만큼 각 교차로의 총통과교통량이 증가한 것으로 나타났다.

〈표 9〉 교차로별 평균대기길이 (단위:m)

구 분	5번 CI			7번 CI			76번 CI		
	기존	개선	증감 (%)	기존	개선	증감 (%)	기존	개선	증감 (%)
북쪽 접근로	-	-	-	191	238	24.6	-	-	-
남쪽 접근로	-	-	-	589	578	-5.8	117	123	4.3
동쪽 접근로	-	-	-	100	97	-3	-	-	-
서쪽 접근로	141	98	-30.5	273	274	0.4	209	217	3.8

〈표 10〉 동서축과 남북축 구간의 평균통행속도 (단위:km/h)

구분	북→남	남→북	동→서	서→동
기존	23.81	18.73	21.19	30.37
개선	29.54	22.64	23.95	28.10
증감	+5.73	+3.91	+2.76	-2.27

하여 분석하였다. center의 로그파일은 검지기가 작동 중일 경우만 생성되므로 3개의 CI교차로만을 대상으로 하여 분석하였다. 통과교통량 분석은 〈표 8〉과 같이 5번 CI는 0.6%, 7번 CI는 2.5%, 76번 CI는 6.8% 만큼 각 교차로의 총통과교통량이 증가한 것으로 나타났다.

구간 운영성과 분석 결과 〈표 10〉과 같이 동서축 방향에서 기존방식과 개선방식의 평균통행속도는 서→동 방향의 경우 2.27km/h 감소하였으며, 동→서 방향의 경우 2.76km/h 증가되었다. 그러나 남북축의 경우는 제어그룹의 지속적인 결합유지로 양 방향 모두 평균통행속도의 향상을 가져왔다.

V. 결론

본 연구에서는 기존의 SA 운영알고리즘의 문제점을 개선하였다. CI가 2개 축 모두의 읍셋 패턴을 계

산하도록 하여, SA내 교차로 운영시 SCI와 MI가 자신의 축 패턴 정보로 운영되도록 하였다. SA 결합/분리시 SA간에 읍셋 패턴 또한 서로 연관성이 있는 패턴을 비교하도록 하였다. 이로써 기존의 한쪽 축 방향으로 SA 운영 및 결합/분리가 이루어지던 것을 동서와 남북 어떠한 축으로도 결합/분리가 가능토록 하였다. 현장실험 결과 개선된 알고리즘과 기존 알고리즘에서의 대기길이는 별 차이가 없었다. 그러나 개선된 알고리즘에서의 통과교통량 및 구간 통행속도가 증가되어 더 효율적인 것으로 분석되었다. 또한 개선된 알고리즘으로 다수의 SA가 결합할 수 있으므로 네트워크 상의 종축과 횡축 방향으로의 보다 효율적인 운영이 가능할 것으로 예상된다.

참고문헌

1. 교통신호제어시스템 기술개발(3차년도) 최종보고서(1993), 서울특별시.
2. 신 전자교통신호시스템 개발 현장종합실험(1995), 서울지방경찰청, 도로교통안전협회.
3. 신신호시스템 평가 및 기능개선 연구(1999), 서울지방경찰청.
4. 2000년 신신호시스템 기능개선 연구(2001), 서울지방경찰청.
5. 2001년 실시간 신호제어시스템 기능개선 용역(2002), 서울지방경찰청.
6. 한동희, 尖端信號시스템의信號制御戰略 평가를 위한 微視的 시뮬레이터의 開發(2002), 서울시립대학교 석사학위 논문.
7. AKCELIK, R., BESLEY M., CHUNG, E., An evaluation of SCATS Master Isolated control (1998).
8. Traffic Control System Handbook(1996), Federal Highway Administration.

- ✉ 주 작 성 자 : 김상운
- ✉ 논문투고일 : 2003. 12. 30
- 논문심사일 : 2004. 3. 2 (1차)
- 2004. 3. 17 (2차)
- 2004. 3. 25 (3차)
- 심사판정일 : 2004. 3. 25
- ✉ 반론접수기한 : 2004. 8. 31