韓國經營科學會誌
第25卷 第1號
2000年 3月

단일 추가제약을 갖는 조합최적화문제를 위한 실용적 완화해법의 계산시간 분석*

홍 성 필**

A complexity analysis of a "pragmatic" relaxation method for the combinatorial optimization with a side constraint*

Sung-Pil Hong**

■ Abstract ■

We perform a computational complexity analysis of a heuristic algorithm proposed in the literature for the combinatorial optimization problems extended with a single side-constraint. This algorithm, although such a view was not given in the original work, is a disguised version of an optimal Lagrangian dual solution technique. It also has been observed to be a very efficient heuristic producing near-optimal solutions for the primal problems in some experiments. Especially, the number of iterations grows sublinearly in terms of the network node size so that the heuristic seems to be particularly suitable for the applications such as routing with semi-real time requirements. The goal of this paper is to establish a polynomial worst-case complexity of the algorithm. In particular, the obtained complexity bound supports the sublinear growth of the required iterations.

1. 서 론

기존의 잘 알려진 조합최적화문제에 단일 제약이 첨가되어 확장된 조합최적화문제는 현대 시스템의 루팅이나 네트워크디자인 문제들에서 많이 발생한다. 지연시간을 제한 받는 최단경로문제[4, 11, 13], 반경(diameter)이나 총 길이를 제한 받는 최소비용신장나무문제[3, 15], 그리고 통신에 참여하는 사용자들 사이에 전송지연시간이 주어진 상한을 넘지 않는 최소비용의 멀티캐스트를 구성하는 문제

* 이 논문은 1999년도 한국학술진흥재단의 연구비에 의하여 연구되었음(KRF-99-E00125).
** 중앙대학교 상경학부

[7, 12], 투어(tour)시간의 제약을 가진 TSP[14] 등이 몇 가지 예이다. 이러한 문제들은 원래 문제 보다 복잡성이 훨씬 증가하여, 대부분 NP-hard이므로 가능한 최적에 근접한 해를 과도하지 않은 계산시간, 즉 다항식시간에 구해내는 것은 최적화분야의 중요한 연구과제이다. 특히 계산시간의 단축은 실제 응용 측면에서는 더욱 강조된다. 한 예로써, 통신 또는 물류망에서의 루팅이나 네트워크디자인은 종종 준실시간(semi-real time)에 가까운 빠른 시간 안에 이루어져야한다. 따라서 어떤 해법이 전체 시스템을 관리 운용하는 체계의 하나의 모듈로써 적절한 효용을 발휘하기 위해서는, 계산시간의 다항식성이 보장되어야하며 더 나아가 해법 구현의 단순성과 투명성이 요구된다.

1995년에 Blokh과 Gutin[6]이 단일 추가제약을 갖는 조합최적화문제를 위해 제안한 해법(이하 BG기법이라 약칭)은 이러한 요건들을 충족하는 것으로 보인다. 서브그래디언트 최적화 기법과 같은 비선형기법에 의존하지 않기 때문에, 해법의 구현이 용이한 장점을 가지고 있다. 특히 알고리듬의 종료조건이 조합최적화기법의 결과에 의해 결정되기 때문에, 서브그래디언트 최적화 기법에서 종종 경험하는 종료조건 설정의 모호함이 없는 장점을 가지고 있다.

특히 원문제의 가능해를 유지해 나가기 때문에 분지한계법에 의존하지 않고도 가능해를 구할 수 있는 장점 또한 가지고 있어 구현상의 용이함이 돋보인다. 보고된 실험 결과에 의하면, 지연시간제약을 가진 최단경로문제[6, 12], 지연시간제약을 가진 스타이너트리문제[7, 12] 등에 적용되어 상당히 빠른 시간 안에 최적해에 근사한 해를 생성해 내는 것으로 밝혀졌다. 특히 [7]과 [12]의 실험결과에 의하면 소요되는 반복단계의 횟수가 바탕(underlying) 네트워크 노드개수의 부선형(sublinear)으로 증가하여, 전체소요시간은 각 반복단계에서 풀게되는 조합최적화문제 알고리듬의 수행시간 보다 크게 증가하지 않는다는 것이다. 따라서 BG기법은 알고리듬 수행시간이 준실시간으로 요구되는 응용문제에 특히 적합한, 실용적 해법으로 보인다.

이러한 장점들에 비해 BG기법의 이론적 연구는 아직 충분하지 않다. 원저자들의 논문에는 알고리듬의 계산시간과 전산실험 결과가 제시되어 있기는 하나, 제시된 계산시간의 상한이 너무 단순하고 문제크기의 다항식시간이 아니며 전산실험 역시 불충분하여, 앞에서 언급한 전산실험에서 관찰할 수 있는 빠른 수행시간, 특히 부선형(sublinear)의 반복횟수를 실험적 또는 이론적으로 설명하기에는 부족하다.

원 논문에서는 이러한 관점을 제시하지 않았지만, BG기법은 라그랑지완화법[8, 10]의 변형이며, 실제로는 단일 승수 라그랑지 쌍대문제의 최적 해법과 동등함(equivalence)을 보일 수 있기 때문에 새로운 알고리듬이라고 볼 수는 없다[1]. 또한 기존에 연구 결과에 의하면 본 논문의 대상문제와 같이 추가제약이 단수가 아닌 일반적인 라그랑지 쌍대문제를 다항식시간에 풀 수 있는 이론적인 해법이 존재한다[5].

그러나 BG기법은 기존의 해법에서 볼 수 없는 새로운 기하학적인 관점을 제시한다. 본 논문에서는 이러한 장점이, 앞에서 언급한 것처럼 원문제를 푸는 효과적인 발견적기법을 제시할 뿐만 아니라, 알고리듬 자체의 이론적 분석을 용이하게 함을 보이려고 한다. 이러한 맥락에서 본 논문은 BG기법의 계산적 복잡성을 분석하려고 한다. 최악의 경우에도 BG기법의 계산시간이 문제입력 크기의 다항식시간으로 제한됨을 증명하는 것이 본 논문의 목표이다. 특히 반복횟수가 [7, 12]의 실험결과에서 나타난 부선형을 설명할 수 있는지가 본 논문의 구체적인 초점이다. 또한 [5]에 제시된 일반 라그랑지 쌍대문제 해법이 단일승수 문제에 적용되었을 때 요구되는 반복횟수와도 비교하려고 한다.

논문의 구성은 다음과 같다. 2절에서는 추가제약을 가진 조합최적화문제를 정의한다. 3절에는 [6]에서 제안한 BG기법을 설명한다. 논문 서술에 있어서 원저자의 논문과는 달리 라그랑지안완화법의 관점에서 서술하여 독자의 이해를 돕고자 한다. 4

절에는 BG기법이 다항식성을 증명한다. 마지막으로 5절에서는 결론과 추후연구방향을 제시하고자 한다.

2. 추가제약을 갖는 조합최적화문제

"단일 추가제약을 갖는 조합최적화문제"란 그 가능한 의미의 범위는 넓으나, 본 논문에서의 의미를 구체화하기 위해 몇 가지 예를 들기로 하자.

$G=(V,E)$를 무방향네트워크라 하자. 최소비용신장나무문제는, 각 호 $e\in E$에 비용 c_e가 주어졌을 때, G의 신장나무 중에서 호비용의 합 $c(T)\equiv\sum_{e\in T}c_e$을 최소로 하는 T를 구하는 문제로, 가장 손쉽게 풀 수 있는 조합최적화문제들 가운데 하나이다. 여기서 T가 만족해야하는 추가제약을 하나 고려해보자. 즉 각 호 $e\in E$에 또 하나의 계수("길이") $d_e\geq 0$을 주고, T의 총길이 $d(T)\equiv\sum_{e\in T}d_e$가 어떤 $\Delta\geq 0$보다 작도록 해를 국한한다. 이 문제를 추가제약을 가진 최소비용신장나무문제(Constrained minimum spanning tree problem) 또는 CMST[3, 15]라고 부르자.

유사한 예를 최단경로문제에 관해서도 만들 수 있다. 각 호 $e\in A$에 비용 c_e가 주어진 유방향네트워크 $N=(V,A)$의 두 마디 s와 t를 연결하는 $s-t$ 경로 중에서, 호비용의 합, $c(P)\equiv\sum_{e\in P}c_e$을 최소로 하는 경로를 구하는 최단경로문제에서 각 호 $e\in E$에 또 하나의 계수("길이") $d_e\geq 0$을 주고, $d(P)\equiv\sum_{e\in P}d_e\leq\Delta$를 만족하는 $s-t$ 경로로 해의 집합을 제한한다. 이러한 문제는 추가제약을 가진 최단경로문제(Restricted shortest path problem) 또는 RSP로 최적화분야에서 비교적 오랫동안 연구되어온[4, 11, 13] 문제이다.

위의 두 예에서 보듯이 본 논문에서 의미하는 "단일추가제약을 가진 조합최적화문제"란 기존의 조합최적화문제에 하나의 제약이 추가되어 확장된 문제를 의미한다. Blokh과 Gutin의 논문에서는, 특히 추가되는 제약이 위의 두 예와 같이, 원래 문제의 목적함수와 같은 종류의 함수에 의하여 정의되는 경우로 문제를 국한하였으며, 이런 문제들을 이모수조합최적화(Combinatorial problems with two parameters)라고 지칭하였다. 그러나 BG기법은 이러한 조건을 만족하지 않는 경우에도 발견적기법을 고안하는데 사용할 수 있다. 예를 들어 반경제약을 가진 스타이너트리문제가 그 예이다[7, 11].

어떤 조합최적화문제에 제약이 한가지 더 추가되는 경우, 확장된 문제는 원래 문제 보다 계산적 복잡도가 훨씬 증가한다. 예를 들어, 다항식시간 해법을 가진 문제 중에서도 가장 쉽게 풀리는 최소신장나무문제와는 달리 CMST는 NP-hard 문제이다[3]. RSP 역시 풀기 NP-hard 문제가 된다[9]. 따라서 이러한 문제들을 위해서는 각 문제의 구조를 탐구하여, 해법의 수행시간을 다항식시간으로 제한하면서도, 근사적인 최적해를 추구하는 근사해법의 개발이 중요한 연구과제가 된다. 예를 들어 두 문제는 모두 NP-hard이기는 하나 또한 소위 "약한 의미"의(weakly) NP-hard로써 해법의 수행시간을 늘이면 원하는 만큼의 오차한계를 보장하는 "polynomial time approximation scheme"이 가능하다[11, 15]. 그러나 이러한 해법들은 "scaling", 부분적인 "enumeration", 병렬연산 등에 의존하는 이론적 해법으로, 현실문제에 실용성은 적다고 하겠다.

3. BG기법

본 절에서는 BG기법을 설명하려고 한다. BG기법은 광범위한 조합최적화문제에 적용될 수 있으나, 서술상의 구체성과 단순성을 위해 본 절에서는 RSP를 중심으로 서술하기로 한다. 또한 모든 표기법과 용어는 2절에서 RSP를 서술할 때 정의한 것들을 사용하기로 한다. 그리고 독자의 편의를 위해, 원저자의 논문과는 달리, 라그랑지안완화법의 관점에서 출발하여 BG기법을 토의하려고 한다.

우리의 문제, RSP는 다음과 같은 수리계획문제 형태로 서술할 수 있다.

$$\min \quad c(P)\equiv\sum_{e\in P}c_e$$

$$\text{(RSP)} \quad \text{s.t } P\text{는 } s-t\text{경로} \qquad \text{(제약식1)}$$
$$d(P) \equiv \sum_{e \in P} d_e \le \Delta \qquad \text{(제약식2)}$$

RSP에서 (제약식2)가 없다면 문제는 기존의 최단경로문제가 된다. 만약 RSP를 라그랑지완화법으로 접근한다면, 가장 자연스러운 쌍대문제는 문제를 어렵게 만드는 제약식(complicating constraint)인 (제약식2)를 비음 라그랑지승수 λ를 사용, 완화하여 얻어진 다음과 같은 형태가 될 것이다(여기서 문제의 요점을 강조하기 위해 목적함수 중 상수항은 제거하였다).

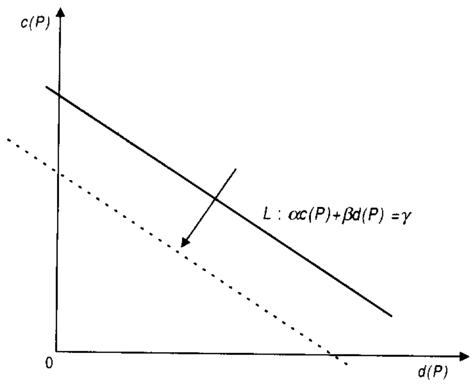
$$\text{(DRSP}(\lambda)) \quad \begin{aligned} z(\lambda) = \ & \min \ c(P) + \lambda d(P) \equiv \sum_{e \in P}(c_e + \lambda d_e) \\ & \text{s.t} \quad P\text{는 } s-t\text{경로} \qquad \text{(제약식1)} \end{aligned}$$

기존 라그랑지완화법은 $\max\{z(\lambda) | \lambda \ge 0\}$을 풀어, (생략된 상수를 더한 후) 원문제의 하한을 구하여, 분지한계법 등에 사용할 것이다[8]. 이에 비하여 BG기법은 다음과 같은 좀더 일반적인 "쌍대문제", SP(α, β)를 사용한다.

$$\text{(SP}(\alpha,\beta)) \quad \begin{aligned} & \min \ \alpha c(P) + \beta d(P) \equiv \sum_{e \in P}(\alpha c_e + \beta d_e) \\ & \text{s.t} \quad P\text{는 } s-t\text{경로} \qquad \text{(제약식1)} \end{aligned}$$

$\alpha \ge 0$와 $\beta \ge 0$를 비용과 지연시간의 비중을 나타내는 모수로 생각하여 그 상대적인 크기를 조절하여 SP(α, β)를 풀면 원문제의 좋은 해가 생성될 수 있다는 완화법의 직관적 측면을 BG기법은 중시한다. 이러한 직관적인 관찰은 기존의 라그랑지안완화법에도 적용되는 것이다. 그러나, BG기법의 경우는 쌍대문제를 최적화하여 좋은 하한을 구하는 것보다는, 매 반복단계에서 승수 α와 β의 조합을 좀더 자유롭게 탐색하여 빠른 시간 안에 좋은 해를 구해내는데 그 요점이 있다고 하겠다.

이러한 BG기법의 직관적 특징을 이해하기 위해, 먼저 위에서 정의한 문제 SP(α, β)의 간단한 기하학적인 의미를 살펴보자. 예를 들어, $\alpha > 0$, $\beta = 0$인 경우 SP(α, β)는 추가제약이 없는 원래 비용 c_e의 최단경로문제가 되며, $\alpha = 0$, $\beta > 0$인 경우는 지연시간 d_e의 최단경로문제가 됨을 쉽게 알 수 있다. 일반적인 경우, 즉 $\alpha > 0$, $\beta > 0$인 경우, SP(α, β)는 각 호의 비용이 c_e와 d_e의 선형조합인 $\alpha c_e + \beta d_e$로 주어진 네트워크의 최단경로문제임을 알 수 있다. 이 때, c_e와 d_e의 선형조합 $\alpha c_e + \beta d_e$를 최소화한다는 것을 [그림 1]과 같이 2차원평면에서 도시할 수 있다. 임의의 $s-t$ 경로 P에 대해, 수평축을 P의 지연시간 $d(P) \equiv \sum_{e \in P} d_e$ 그리고 수직축을 비용 $c(P) \equiv \sum_{e \in P} c_e$을 나타내는 좌표축으로 생각하면, 모든 $s-t$ 경로는 이 2차원평면의 한 점으로 표시 할 수 있다.



[그림 1] BG기법의 기하학적 의미

이때 주어진 α와 β의 값에 대하여 $f(P) \equiv \alpha c(P) + \beta d(P)$가 어떤 동일한 값 γ가 되는 경로들의 집합은, 그림에서 실선으로 표시된 어떤 직선 L 위에 포함 될 것이다. 또한 직선 L과 평행인 직선 위의 경로들은 모두 동일한 $f(\cdot)$값을 가지며 직선이 원점과 멀어질수록 그 값은 커진다. 따라서 SP(α, β) 문제는 직선 L과 수직을 이루는 방향으로 L을 원점과 최대한 가깝도록 평행이동 시켜 얻어진 직선 위의 $s-t$경로를 구해내는 문제임을 알 수 있다. 특별한 경우로, SP(1,0)는 수평선을 수직 아래방향으로, SP(0,1)는 수직선을 수평 좌방향으로 최대한 평행이동 하는 의미로 해석할 수 있다. 이러한 기하학적인 관점은 다음에 서술된 BG기법을 이해하는데 도움이 된다.

BG기법

단계 1 : 최단경로알고리듬을 사용하여 SP(1,0), 즉 비용 c_e의 최단경로문제를 풀어 얻어진 경로를 Q라고 하자. 만일 $d(Q)$가 $d(Q) \le \Delta$를 만족하면 Q는 최적해이므로 알고리듬을 종료한다.

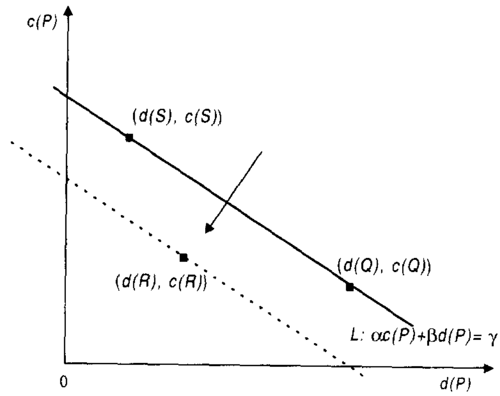
단계 2 : 최단경로알고리듬을 사용하여 지연시간 d_e의 최단경로문제, SP(0,1)을 풀어 경로 S를 구한다. 만약 $d(S) > \Delta$가 성립하면 가능해는 존재할 수 없으므로 알고리듬을 종료한다.

단계 3 : $\alpha \leftarrow d(Q)-d(S)$, $\beta \leftarrow c(S)-c(Q)$, 그리고 $\gamma \leftarrow d(Q)c(S)-d(s)c(Q)$로 정의하고, 각 호의 비용을 $f_e \equiv \alpha c_e + \beta d_e$로 놓는다. 그리고 최단경로알고리듬을 사용하여 f_e를 호 비용으로 하는 최단경로문제를 풀어 경로 R을 구한다.

단계 4 : 만약 $f(R) \equiv \sum_{e \in R} f_e = \gamma$이고, $d(R) \le \Delta$이면 R을 RSP의 해로 출력한다. 만약 $f(R) = \gamma$이고, $d(R) > \Delta$이면 S을 RSP의 해로 출력한다.

단계 5 : 만약 $f(R) < \gamma$이고, $d(R) \le \Delta$ 이면 $S \leftarrow R$로 놓는다. 만약 $f(R) < \gamma$이고, $d(R) > \Delta$이면 $Q \leftarrow R$로 놓는다. 단계 3으로 간다.

초기화인 단계 1과 단계 2의 의미는 자명하므로 설명을 생략한다. 단계 3에서는 $f(P) = \alpha c(P) + \beta d(P)$을 최소화하는 $s-t$경로 P를 구하는 최단경로문제를 풀게 되는데, 이 때 정의된 식에 의해서 α와 β의 값을 넣으면, 평행이동의 기준이 되는 직선 L은 Q와 S에 대응되는 두 좌표인 ($d(Q)$, $c(Q)$) 그리고 ($d(S), c(S)$)을 지나는 직선이 됨을 쉽게 보일 수 있다 ([그림 2] 참조). 또한 주어진 γ는 (Q와 S를 포함한) 직선 L 상의 경로들이 갖게되는 동일한 $f(\cdot)$값이 됨도 쉽게 확인할 수 있다. 따라서 단계 3에서는 직선 L에 수직이며 원점과 가까운 방향으로 최적화하여 경로 R을 구하게 된다.

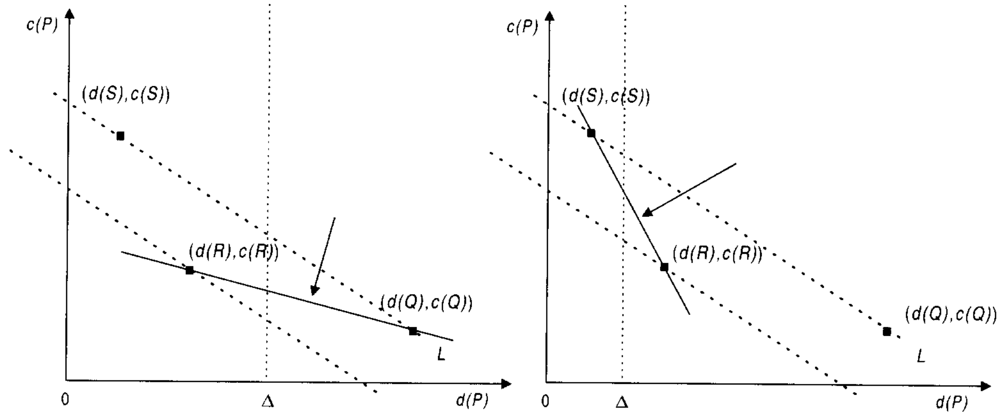


[그림 2] BG기법의 초기 단계

단계 4에서는, 단계 3에서 얻어진 R의 목적함수값 $f(R)$이 실제 개선이 없어, Q와 S의 $f(\cdot)$값인 γ와 동일한 경우 알고리듬을 종료한다. 이때 R이 추가제약 조건 $d(R) \le \Delta$을 만족하면 R을 RSP의 해로 출력한다. 그렇지 않은 경우, 현재 추가제약조건을 만족하는 해 중에 가장 비용이 작은 해인 S가 발견적 해로 출력된다.

단계 5에서는, 단계 3에서 얻어진 R의 목적함수값 $f(R)$이 Q와 S의 $f(\cdot)$값인 γ보다 개선이 된 경우 새로운 최적화의 방향을 결정한다. 만일 R이 추가제약 조건 $d(R) \le \Delta$을 만족하면 R로 S를 대체한 후 단계 3에서 새로운 반복단계가 시작된다. 이 것은 앞에서 설명한 것처럼, 평면에서 Q와 R을 지나는 직선과 수직인 방향으로 최적화하는 것을 의미한다 [그림 3(a)]. 그렇지 않은 경우, R로 Q를 대체한다. 이 것은 평면에서 R와 S을 지나는 직선과 수직인 방향으로 최적화하는 것을 의미한다[그림 3(b)].

이렇게 단계 3에서 새롭게 시작되는 반복단계의 의미는 [그림 3(a)]와 [그림 3(b)]를 비교하면 명확해진다. 앞에서 언급했듯이 최적화의 방향이 수직아래 방향이면 비용을 최소화하는 최단경로를, 그리고 수평 좌방향이면 지연시간을 최소화하는 최단경로를 구하는 문제임을 상기하자. [그림 3(a)]의 경우는 직전단계에서 구한 경로 R이 추가제약(지연시간)조건을 만족하는 경우이며, [그림 3(b)]의



[그림 3(a)] [그림 3(b)]

[그림 3] BG기법의 일반 단계

경우는 그렇지 못한 경우이다. 이때 최적화방향을 나타내는 화살표의 방향을 비교하면 전자의 경우가 더 수직 아래 방향과 가깝다는 것을 알 수 있다. 즉, 직전 단계에 구해진 R이 가능해인 경우는 새로운 반복단계에서, 그렇지 않은 경우보다 비용 최적화 쪽에 상대적으로 더 큰 가중치를 두어 새로운 경로를 추구하게된다. 물론 이러한 관점은 대칭적으로, 직전단계에 가능해가 구해지지 못한 경우는 새로운 반복단계에서 지연시간 최소화에 더 큰 비중을 두어 새로운 경로를 추구하는 것으로 해석할 수 있다.

요약하면, BG기법은 매 단계마다 세 개의 경로를 유지하는 데, 첫째는 현재까지 구한 경로들 중에서, 지연시간을 만족하면서 비용이 최소인 경로 S, 둘째는 현재까지 구한 비가능해 중에서 지연시간이 가장 작은 경로 Q, 그리고 셋째는 Q와 S의 상대적인 위치가 결정하는 방향으로 최적화하여 얻어진 경로 R을 말한다. 이때 R의 가능성 여부에 따라 비용 또는 지연시간에 최소화에 더 큰 가중치를 둔 최적화를 수행하며, 개선이 가능한 한 반복된다.

본 절에서 살펴 본 BG기법에서 앞 절에서 언급한 실용적인 강점들을 확인할 수 있다. 우선 기법의 승수 수정이 매우 직관적이라는 것이다. 매 반복단계에서 동일한 조합최적화알고리듬을 사용하기 때문에 구현이 용이하다. 알고리듬의 종료조건이 비선형기법에 의존하지 않고 조합최적화 알고리듬에서 생성된 해의 값에 의존하기 때문에 해법의 적용이 투명해진다는 것이다. 모든 반복단계에서 그 때 까지 가장 좋은 가능해(S)를 유지하기 때문에 별도의 알고리듬이 없이도 가능해를 확보할 수 있다. 그리고 승수의 수정이 3개의 연속한 해, Q와 S 그리고 R에 의해 결정되기 때문에 해의 질이 높을 개연성이 크다.

또한 BG기법은 넓은 적용성을 갖고 있다. RSP를 중심으로 기술된 BG기법에서는 각 반복단계에서의 $f(\cdot)$ 최적화는 바로 하나의 최단경로문제에 해당한다. CMST에 BG기법을 적용하면 한 반복단계는 최소신장나무문제를 푸는 것이 될 것이다. 이렇듯 추가제약식과 목적함수가 같은 종류일 때는 BG기법은 모든 단계에서 추가제약이 없는 원래 조합최적화문제를 풀게되며, 따라서 이러한 문제를 푸는 (최적알고리듬이 아니더라도) 효율적인 알고리듬이 존재하기만 하면 BG기법을 적용할 수 있다. 더 나아가 추가제약과 목적함수가 같은 종류가 아니더라도, 단계 3에서 정의된, 두 종류의 계수의 양의 선형조합으로 정의된 문제를 효과적으로 풀 수 있다면 BG기법은 이를 매 반복단계에서 서브루틴으로 사용하여 전체문제를 풀어나갈 수 있다.

마지막으로 특기할 사항은 BG기법은 실제로 라그랑지 쌍대문제를 최적화하는 어떤 기법과 동등하다는 점이다[1]. 이 기법은 승수가 단수일 때, 라그랑지 쌍대문제는 변수가 한 개인 부분별 직선함수(piecewise linear function)이라는 점에서 자연스럽게 고안될 수 있는 최대 값을 찾는 해법이다. 본 논문의 초점과는 벗어나기 때문에 이 해법의 논의는 생략한다. 그러나 언급할 점은 이러한 해법의 다항식성이나 그 복잡성은 저자가 아는 바로는 아직 규명된바 없으며, 따라서 다음절에 제시될 BG기법의 다항식성과 복잡성은 이 기존 해법의 최초의 다항식성 증명이라고 할 수 있다.

4. BG기법의 다항식성

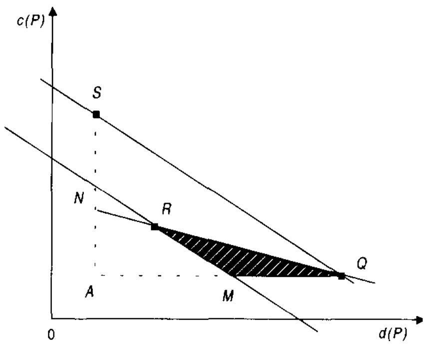
본 절에서는 BG기법의 계산시간이 최악의 경우에도 문제의 입력 크기의 다항식으로 주어짐을 증명하고자 한다. 먼저 쉽게 구할 수 있는 상한을 살펴보자. 초기화 단계를 제외한 매 반복단계(단계 3 - 단계 5)에서 구해지는 경로 R은 주어진 양수 α와 β의 값에 대하여 $f(R)=\alpha c(R)+\beta d(R)$을 최소화하는 경로가 된다. 따라서 $s-t$ 경로 중에는 R보다 비용과 지연시간이 모두 작은 경로란 존재하지 않는다. 만일 존재한다면 R의 $f(\cdot)$에 대한 최적성에 모순된다. 따라서 단계 3에서 구해지는 경로 R은 비용과 지연시간의 두 개의 목적함수의 관점에서 볼 때, 파레토최적(Pareto optimal) 경로에 해당한다. 또한 같은 경로가 단계 3에서 2회 이상 반복해서 구해지는 일이 발생할 수 없음을 증명할 수 있다. 따라서 전체 계산시간은 전체 파레토최적경로의 개수에 최단경로알고리듬의 수행시간을 곱한 것이 된다. 이상은 원저자의 논문[6]에 주어진 계산시간분석이다. 그러나 파레토최적경로의 개수가 문제입력크기의 다항식인지 여부가 알려져 있지 않아, 지금까지는 BG기법의 다항식성은 긍정 또는 부정적 방향 어느 쪽으로도 규명된 바 없다.

본 논문에서 증명할 BG기법의 다항식성은 다음과 같은 관찰에서 출발한다. [그림 4]를 보자 (앞의 그림들과는 달리 표기상의 단순성을 위하여 각 경로에 대응하는 점을 좌표로 표시하는 대신 경로의 명칭으로 표기하였다). 초기화인 단계 1과 2를 수행하여 Q와 S를 구하면, 찾아야할 경로는 삼각형 AQS 안에 국한됨을 알 수 있다. 단계 3에서 직선 QS와 수직인 방향으로 최적화하여 R을 구하면 알고리듬은 단계 4에서 종료되거나, 그렇지 않은 경우, 단계 5로 가게된다. 이 경우 [그림 4]와 같이 R이 위치한다고 하면, 모든 경로는 사각형 $NMQS$에 존재함을 의미한다.

다음 반복단계에서는 R의 가능성, 즉 지연시간 만족 여부에 따라, 직선 QR 혹은 직선 RS와 수직인 방향으로 최적화를 하여 단계 3을 다시 시작하게 된다. 전자의 경우, 즉 R이 지연시간 제약을 만족하여, 직선 QR에 수직인 방향으로 최적화하게 되면, 그 이후에 알고리듬이 단계 3에서 생성하는 경로들은 모두 삼각형 MQR에 국한된다. 따라서 최초의 R이 구해진 시점에서 두 번째 R을 구하기 시작한 시점까지 탐색영역은 사각형 $NMQS$에서 삼각형 MQR로 축소되었음을 알 수 있다. 그리고 삼각형 MQR의 면적은 사각형 $NMQS$의 면적의 반보다도 작음을 알 수 있다. 한편, R이 지연시간 제약을 만족하지 않는 경우에 대해서도 유사하게 탐색영역의 면적이 반 이상 줄어듦을 알 수 있다. 이러한 현상은 반복된다. 즉 두 번째 R을 구한 후 세 번째 R을 구하기 직전까지 탐색영역은 삼각형 MQR에 포함된 어떤 사각형으로부터, 사각형의 면적의 반보다 작은 면적의 삼각형으로 줄어드는 것을 귀납적으로 쉽게 알 수 있다. 따라서 반복단계의 횟수가 하나씩 증가할 때마다, 탐색영역은 1/2 이상 감소하는 것을 알 수 있다. 이러한 관찰과 다음과 같은 잘 알려진 정리[16]를 결합하면 BG기법의 다항식성을 유도할 수 있게된다.

정리 1 [16] : 주어진 행렬, $A \in R^{m \times n}$에 대하여, 다면체 $\Pi=\{x|Ax \le b\}$가 유한하며 (무한 반직선을 포함하지 않으며) 내부점을 갖고 있어 차원이 n이라고 하

자. 또한 A나 b의 절대값이 가장 큰 원소의 이진법 입력길이를 μ라고 하자. 그렇다면 다면체 Π의 n차원 부피는 최소한 $2^{-4n^3\mu}$가 된다.



[그림 4] 탐색 영역 감소 과정

[그림 4]에서 알 수 있는 것처럼, 매 반복단계에서 R을 구하기 직전에 남은 삼각형의 탐색영역은 다음의 세 개의 반공간(halfspace)의 공통영역으로 주어진다.

$$\begin{aligned} \alpha_1 x+\beta_1 y &\geq \gamma_1 \leftrightarrow \text{직선MQ} \\ \alpha_2 x+\beta_2 y &\geq \gamma_2 \leftrightarrow \text{직선MR} \qquad \text{(식 4.1)} \\ \alpha_3 x+\beta_3 y &\leq \gamma_3 \leftrightarrow \text{직선RQ} \end{aligned}$$

여기서 계수들은 모두 비음이며, 어떤 반복단계에서도 반공간을 결정하는 2차원 초평면, 즉 직선들은 단계 4에서 $f(\cdot)$가 개선되지 않아 알고리듬이 종료되기 전까지는, 서로 겹치거나 평행해지는 경우가 없다는 것을 쉽게 알 수 있다. 그러므로 알고리듬 종료 전까지 삼각형은 내부점을 갖게되며, 따라서 정리 1에 의하여, $n=3$이므로 삼각형의 면적은 $2^{-36\mu}$보다 작아지는 일은 생길 수 없음 알 수 있다. 여기서 μ는 식 (4.1)의 계수 중에서 가장 큰 수의 로그 연산을 취한 값을 의미한다. 그런데, BG기법에서 이러한 계수들은, 각 반복단계에서 구한 Q와 S에 대하여, $c(S)-c(Q)$, $d(Q)-d(S)$ 또는 $d(Q)c(S)-d(S)c(Q)$로 주어짐을 보았다. 따라서 비용계수나 지연시간 계수 중에 가장 절대값이 큰 수의 절대값을 C라고 하면, $c(Q)$, $d(Q)$, $c(S)$, 그리고 $d(S)$들의 절대값은 모두 nC를 넘지 않으므로, μ는 $\log nC$의 어떤 상수 k배를 넘지 않는다. 따라서 탐색영역은 삼각형AQS에서 시작하여 한 번의 반복단계가 끝날 때마다 반 이상 줄지만, 알고리듬 종료 전까지 $2^{-k\log nC}$보다 작아지는 일이 발생하지 않는다. 따라서 반복단계의 횟수는 다음의 수보다 커질 수 없다.

$$\begin{aligned} &\log(\text{ AQS의면적}/2^{-k\log nC}) \\ =\ &\log(\text{ AQS의면적})+k\log nC \\ =\ &\log(d(Q)-d(S))(c(S)-c(Q))+k\log n \\ &+k\log C \leq \log d(Q)c(S)+k\log n+k\log C \\ &\leq 2\log nC+k\log n+k\log C \\ =\ &(k+2)(\log n+\log C) \\ =\ &O(\max\{\log n, \log C\}) \end{aligned}$$

따라서 다음과 같은 정리가 얻어진다.

정리 2: BG기법을 RSP에 적용할 때, 알고리듬 계산시간은 다음과 같다.

$$O(\text{최단경로해법 계산시간} \times \max\{\log n, \log C\}) \qquad \text{식 (4.2)}$$

단, C는 비용 및 지연시간 계수 중 가장 큰 절대값을 갖는 원소의 절대값을 의미한다.

정리 2에 얻어진 결과는 다른 문제들, 예를 들어 CMST에도 적용될 수 있음을 쉽게 알 수 있다. BG기법의 반복횟수는 노드 n과 최대입력크기 계수의 로그함수가 됨을 알 수 있다. 따라서 계수들이 스케일이 비슷하게 유지된다면 네트워크의 크기가 증가해도 반복횟수의 증가속도는 그보다 훨씬 작은 증가를 보이며, 이는 다른 실험적 연구에서 관찰된 부선형 현상들을 설명해주는 것이라고

할 수 있다.

앞에서 언급한 바와 같이 BG기법은 원문제를 푸는 발견적기법일 뿐 아니라, 라그랑지 쌍대문제의 최적해법이기도 하다. [5]에 주어진 일반적 라그랑지 쌍대문제의 해법을 단일추가제약문제의 쌍대문제에 적용했을 때, 계산되는 반복횟수는 O (문제입력크기)로 주어진다. 따라서 이러한 시간을 그대로 해석하면, 본 논문의 문제들을 위한 경우, BG기법의 수행시간이 더 좋다고 말할 수 있겠다. 물론 [5]의 해법도 단일추가제약 조합최적화 문제에 맞게 기술된다면 본 논문에서 얻어진 반복횟수 정도로 개선될 가능성도 있다. 어쨌든 BG기법은 현재까지 문헌에서 알려져 있는 단일추가제약 조합최적화문제를 위한 가장 빠른 기법 중에 하나라고 할 수 있다.

5. 맺는 말 그리고 연구방향

본 논문에서는 단일추가제약으로 확장된 조합최적화문제에 광범위한 적용성을 갖는 실용적인 발견적기법의 복잡성을 논의하였다. 이 기법은 실험적 연구에서 좋은 해를 상당히 빠른 시간 안에 구하는 것으로 관찰되었는데, 본 논문에서는 이론적으로 그 현상을 규명하였다. 특히 반복단계의 횟수가 문제의 계수들의 크기가 심하게 증가하지 않는 경우, 노드개수의 로그함수이므로 실제 알고리듬의 수행시간은 각 반복단계에서 사용하는 조합최적화 알고리듬의 효율성에 의해 결정되어, 기법의 실용성이 이론적으로 확인된 것이라고 하겠다. 또한 이 기법은 원문제를 위한 발견적기법일 뿐 아니라, 라그랑지 쌍대문제를 위한 가장 빠른 최적 해법 중 하나라고 할 수 있다.

마지막으로, 토의한 발견적기법이 기하학적으로 상당히 직관적인 특징을 가지고 있어서 더 다양한 변형 알고리듬이 가능하다는 것이다. 예를 들어 승수를 수정할 때, 현재 해들의 상대적 위치를 다른 방법으로 계산된 가중치로 반영하는 변형 기법의 실험적 연구도 가능할 것이다.

감사의 글

BG기법이 실제로 단일 승수 라그랑지 쌍대문제를 최적으로 푸는 알고리듬과 동등하다는 것[1]과 일반적인 라그랑지 쌍대문제의 다항식해법에 관한 참고문헌[5]을 지적해주신 단국대학교의 명영수 교수님께 감사를 드립니다.

참 고 문 헌

[1] 명영수, 「Personal communication」, 1999.

[2] 홍성필, 민대현 "추가제약을 가진 MST문제를 위한 실용적 근사해법", 「한국경영과학회 98 추계학술대회발표논문집」, pp.275-277.

[3] Aggarwal, V., Y.P. Aneja, and K.P.K. Nair, "Minimal Spanning Tree Subject to a Side Constraints", *Compt. Oper. Res.*, Vol.9 (1982), pp.287-296.

[4] Aneja, Y.P., V. Aggarwal, and K.P.K. Nair, "Shortest Chain Subject to Side Constraints", *Networks*, Vol.13 (1983), pp.295-302.

[5] Bertsimas, D. and J.B. Orlin, "A Technique for Speeding Up the Solution of the Lagrangian Dual", *Mathemaical Programming*, Vol.63 (1904), pp.23-45.

[6] Blokh, D. and G. Gutin, "An Approximation Algorithm for Combinatorial Optimization Problems with Two Parameters", Technical Report Preprint-95-14.ps, Dept. Math. Compt. Sci., Odense University, 1995 (also available from www.imada.ou.dk/Research/ Preprints).

[7] Chung, S.J., S.-P. Hong, and H.-S. Huh, "An Efficient Multicast Routing Algorithm for Delay-Sensitive Applications", in *Proc. 1997 IEEE GLOBECOM*, pp.1898-1902.

[8] Fisher, M.L., "The Lagrangean Method for Solving Integer Programming Problems", *Management Sci.*, Vol.27 (1981), pp.1-18.

[9] Garey, M.R. and D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.

[10] Geoffrion, A.M., "Lagrangean Relaxation for Integer Programming", *Math. Programming Stud.*, Vol.2 (1974), pp.82-114.

[11] Hassin, R., "Approximation Schemes for the Restricted Shortest Path Problem", *Math. Oper. Res.*, Vol.17 (1992), pp.36-42.

[12] Hong, S.-P., H. Lee, and B. H. Park, "An Efficient Multicast Routing Algorithm for Delay-Sensitive Applications with Dynamic Membership", in *Proc. 1998 IEEE INFOCOM.*

[13] Joksch, H.C., "The Shortest Route Problem with Constraints", *J. Math. Anal. Appl.*, Vol.14 (1966), pp.191-197.

[14] Melamed, I.I., S.I. Sergeev, and I.Kh. Sigal, "The Traveling Salesman Problem: Part 1, Theoretical Issues", *Automation and Remote Control*, Vol.50 (1989), pp.1147-1173.

[15] Ravi, R. and M.X. Goemans, "The Constrained Minimum Spanning Tree Problem", An Extended Abstract, 1995.

[16] Schrijver, A., *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.