대형 혼합 정수계획법 프로그램 MIPBB의 개발†

Developing a Large-Scale Mixed Integer Programming Program MIPBB

박순달*, 도승용**, 이상욱*, 이태호***, 황성섭*

ABSTRACT

The purpose of this paper is to develope a large-scale mixed integer program MIPBB. In this paper, the various issues such as branching strategies, searching and bounding strategies, storing basis information, handling numerical instability, that are important for developing a large-scale mixed integer programming program, are considered. And the experimental results of MIPBB are presented and compared to those of GLPK.

Keywords : Linear Programming, Integer Programming, Branch and Bound.

1. 서론

혼합정수계획법(Mixed Integer Programming)은 목적함수와 제약식이 일차식이고, 변수들이 정수 또는 실수를 가지는 수리계획법을 의미한다[1]. 혼합정수계획법은 생산 및 제조 시스템, 정보/통신, 물류 및 유통 시스템 등 많은 분야에 응용되고 있는 수리계획법의 한 분야이지만, 최적해를 구하는 데 매우 많은 시간이 소요될 뿐 아니라, 현실에서 발생하는 문제는 대형인 경우가 많다. 관련 프로그램으로는 CPLEX[6], MINTO[4], bc-opt[3], ABACUS[7], XPRESS[15], MOSEK[14], GLPK[13] 등이 있지만 대부분 외국에서 개발되어진 상용 소프트웨어이며, 국내의 프로그램은 교육용 또는 소규모의 문제를 풀 수 있는 수준이다.

혼합정수계획법의 대표적인 해법으로는 분지한계법(Branch and Bound)과 절단평면법(Cutting Plane)이 있다. 최근에 효율적인 해법으로 알려져 있는 분지절단법(Branch-and-Cut)은 분지한계법의 기반하에 절단평면을 첨가하여 해를 구하는 방법이다[2]. 이러한 해법들이 현실의 대형문제를 위해 구현되어질 때 중요한 요소는 정수변수의 개수가 많아질수록 급격히 증가하는 부문제의 수와 이를 저장하기 위한 메모리 공간의 크기이기 때문에 부문제의 수를 줄이는 전략들과 부문제들에 관련된 정보를 효율적으로 저장, 관리할 수 있는 자료구조의 사용이 필수적이다. 또한, 분지전략, 탐색전략, 한계전략, 부문제 관련정보의 보관방법, 수치적으로 불안정한 부문제의 대처방법 등이 중요하다.

따라서 본 연구에서는 대형 정수계획법 프로그램의 구현 시에 고려해야 될 사항과 문제점을 고찰하고 이를 대형정수계획법 프로그램 MIPBB에서 해결하는 방법을 제시하도록 한다.

2. 분지한계법

혼합정수계획법을 수식으로 표현하면 다음과 같다.

$$
\begin{aligned}
max \quad z_{MIP} &= \sum_{j\in I} c_j x_j + \sum_{j\in C} c_j x_j \\
s.t. \quad \sum_{j\in I} c_j x_j &+ \sum_{j\in C} c_j x_j \le b_i, \qquad i=1,\cdots,m \\
l_j \le x_j &\le u_j \qquad j\in N \\
x_j &\in Z \qquad j\in I \qquad (2.1) \\
x_j &\in R \qquad j\in C
\end{aligned}
$$

여기서, I는 정수변수의 집합이고, C는 비정수변수의 집합 , N은 I와 C의 합집합이다.

분지한계법의 절차와 본 연구에서 대상으로 하는 부분들을 나타내보면 다음의 <표 1>과 같다.

< 1> 분지한계법 및 전략

단계	내용	설 명	관련전략
단계 1	초기화	식(2.1) · 선형계획법으로 푼다. 비가해이거나 정수해이면 끝난다.	
단계 2	분지	분지변수를 선택하여 원문제에, 분지된 두 개의 새로운 문제를 문제목록에 등록한다.	분지전략, 부문제 저장
단계 3	탐색	문제목록에서 문제 하나를 선택한다. 만일 비어 있으면 끝난다.	탐색전략
단계 4	한계	선택된 문제를 풀고 개선여부를 판별한 후, 비정수해이고 현재의 정수가능해보다 크면 단계 2 나머지는 단계 3 .	원단체법과 쌍대단체법의 이용, 한계전략

3. 분지전략

정수계획법에서는 정수변수들 중에서 실수변수의 값을 가지는 변수를 선택하고 새로운 한계를 주어서 문제의 공간을 나누게 된다. 선형계획법으로 이완(이하 LP이완)된 부문제를 풀면 정수변수들 중에서 실수값을 가지는 정수변수가 존재하게 되고 그 중 하나를 선택한다. 분지한계법을 효과적 수행을 위해서는 분지 변수 선택 시 목적함수값의 상하한을 빠르게 수렴시키는 변수 선택은 부모문제와 자식문제의 목적함수값의 차이를 크게하는 변수의 선택이 유리하다.

최대 정수 비가능(Maximum Integer Infeasibility)을 이용하는 방법

$f_i = \bar{b}_i - \lfloor \bar{b}_i \rfloor$일 때, 실수값을 가지는 정수변수들 중에서 $\max_{i\in I} \min \{f_i, 1-f_i\}$인 변수 x_i를 분지할 변수로 선택하는 방법이며, 정수의 정도가 가장 좋지 않은 변수를 분지변수로 선택하는 것이다.

패널티(Penalty)를 이용하는 방법

최적해의 상하한의 수렴을 위하여 일반적으로 부모문제와 자식문제간의 목적함수값 차이를 크게 하는 변수를 선택하는 것이 유리하며, 실수값을 갖는 정수변수에 상하한을 주면 원비가능이 발생한다. 쌍대단체법을 이용 원가능을 회복하기 위해서는 선회연산을 실시하며, 1회 선회연산

† 서울대 발전기금 일반학술 연구비 지원 연구과제 (4000-2002-0081) .

* 서울대학교 산업공학과
** 삼성전자
*** 한국신용정보

을 실시했을 때의 목적함수값의 감소량을 이용하여 분지변수를 선택하는 방법이다[10]. 변수 x_i를 실수값을 갖고 있는 정수변수라고 가정하면

$$x_i = \overline{b}_i - \sum_j \overline{a}_{ij}(x_j) \tag{3.1}$$

$x_i \le \lfloor \overline{b}_i \rfloor$과 $x_i \ge \lceil \overline{b}_i \rceil$를 제약식에 추가하여 x_i를 분지하고 표준형으로 만들면 다음과 같다.

$$x_i + x_i^D = \lfloor \overline{b}_i \rfloor, \quad x_i - x_i^U = \lceil \overline{b}_i \rceil \tag{3.2}$$

식 (3.1)을 식 (3.2)에 대입하면

$$\begin{aligned} x_i^D &= (\lfloor \overline{b}_i \rfloor - \overline{b}_i) + \sum_j \overline{a}_{ij} x_j \\ x_i^U &= (\overline{b}_i - \lceil \overline{b}_i \rceil) - \sum_j \overline{a}_{ij} x_j \end{aligned} \tag{3.3}$$

이 된다. 여기서 x_i^D, x_i^U가 음수이므로 원가능성이 깨지게 되고, 쌍대단체법에서 x_i^D, x_i^U이 1회 선회연산에서 탈락변수가 된다. $x_i \le \lfloor \overline{b}_i \rfloor$을 추가하고 1회 선회연산을 실시했을 때, 목적 함수값의 감소량을 다운 패널티(Down Penalty)라 한다. $x_i \ge \lceil \overline{b}_i \rceil$인 경우 업 패널티(Up Penalty)라 하며 식(3.4)와 같이 계산이 된다.

$$P_D = \min\left\{ -\frac{(\lfloor \overline{b}_i \rfloor - \overline{b}_i)}{\overline{a}_{ij}} \overline{c}_j : (\overline{a}_{ij} > 0,\ j \in R) \text{ 또는 } (\overline{a}_{ij} < 0,\ j \in R') \right\}$$

$$P_U = \min\left\{ \frac{(\overline{b}_i - \lceil \overline{b}_i \rceil)}{\overline{a}_{ij}} \overline{c}_j : (\overline{a}_{ij} < 0,\ j \in R) \text{ 또는 } (\overline{a}_{ij} > 0, j \in R') \right\} \tag{3.4}$$

(단, R:하한 비기저 지수, R':상한 비기저 지수)

패널티를 계산에 많은 시간이 소요되지 않으며, 분지변수 선택 시에 계산되어진 패널티 값을 이용하여 LP이완된 부문제의 목적함수값에 대한 상한(Upper Bound)을 강화시킬 수 있는 장점이 있으므로 효율적이다.

4. 탐색전략

탐색전략의 목적은 좋은 정수가능해를 빨리 찾아 현재의 정수가능해보다 더 좋은 정수가능해가 존재하지 않음을 보이는 것이다. 탐색전략은 깊이우선탐색방법, Best-First탐색방법, Best-Projection탐색방법 등이 있다.

깊이우선 탐색방법

문제 목록에서 가장 최근에 생성된 문제를 선택하는 방법이다. 깊이가 늘어날수록 제약이 많아져 정수해에 근접할 가능성이 뿌리노드(root node)보다 많아서 탐색에 필요한 기억공간의 요구량이 적고 정수해를 보다 빨리 찾을 가능성이 많다. 또한 부모문제의 최적기저를 이용하여 자식문제에 쌍대단체법을 적용할 수 있다는 장점이 있지만 목적함수 값의 개선속도가 느리다.

Best-First 탐색방법

문제 목록에서 문제 선택 시 목적함수 값의 한계가 가장 좋은 문제를 먼저 선택하는 방법이며, 정수해를 찾았을 경우 최적해에 가깝다. 이 방법은 정수 가능해가 빨리 발견되지 않는 경우, 풀어야 하는 부문제의 수가 크게 증가하여 많은 메모리를 요구하게 된다. 또한, 이전 문제와 다음 문제가 서로 연관성이 떨어지게 되어 부문제를 푸는 경우 많은 계산량을 요구하게 된다. 이러한 단점은 문제목록의 문제들의 부모문제의 기저정보를 저장한 후 이 정보를 이용함으로서 극복할 수 있다.

Best Projection 탐색방법

문제 목록의 문제가 정수해를 가졌을 때의 목적함수값을 추정하여 추정값이 가장 좋은 문제를 선택한다.

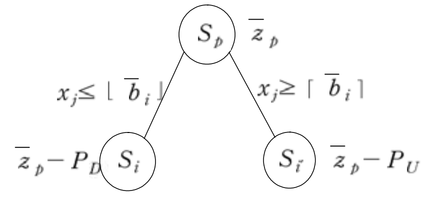
$$E_i = \overline{z}_i + \left(\frac{z^t - \overline{z}_0}{s_0} \right) s_i, \quad i \in N \tag{4.1}$$

단, $s_i = \sum_{j \in I} \min(f_j, 1 - f_j)$, $\overline{z}_i$는 부문제 i의 상한, $\overline{z}_0$는 루트노드의 상한, s_0는 루트노드의 정수비가능의 합.

이 방법은 문제를 풀기 전에 정수 가능해를 가졌을 때의 목적함수값을 예측할 수 있게 한다. 정수가능해를 찾은 후에는 식 (4.1)를 이용하여 탐색 대상 부문제들의 목적함수값을 추정하고 값이 가장 좋은 문제를 선택한다. 이것은 좋은 정수 가능해를 찾을 가능성을 높이며 좋은 정수해를 찾은 경우 많은 부문제들을 분지끝이 가능하게 됨으로써 전체 수행속도에 성능향상을 유도한다. 부문제 i의 목적함수 값으로 사용할 수 있는 값은 MIPBB에서는 다음의 두 가지 방법을 사용하였다.

- 부모문제의 LP이완 목적함수 값을 이용
- 패널티값을 이용하여 목적함수 값의 한계강화

부문제 i와 부문제 p의 LP이완 목적함수 값을 z_i, z_p라고 하면 $z_p \ge z_i$이므로 z_p는 부문제 i에 대해 상한을 제공한다. 따라서 부문제 i의 부모문제의 LP이완 목적함수 값을 부문제 i의 상한으로 사용할 수 있다.



[1] 패널티 값을 이용한 부문제의 상항

패널티 값을 이용하여 목적함수 값의 한계를 강화시키는 방법은 [그림 1]과 같다. 부모문제 p를 LP이완하여 풀었을 경우의 목적함수 값을 $\overline{z}_p$라고 하면 부문제 i와 i'은 분지과정을 통해 생성되어진다. 분지변수 선택 시에 업 패널티 값과 다운 패널티값을 계산하게 된다. 이것을 분지변수 선택시의 기준으로 사용할 뿐만 아니라 생성되어지는 부문제 i와 i'의 상한으로 사용할 수 있다.

5. 한계전략

한계전략은 문제를 분지끝(Fathomed)시키는 전략이다.

부 문제 풀이 중 강제종료시키는 방법

LP이완 부문제를 풀 때, 현재 정수가능해의 목적함수값보다 작은 값을 가지는 경우, 분지끝 시킬 수 있다.

비가능 판정

만일 패널티를 구할 수 없는 경우, LP이완된 부문제를 쌍대 단체법으로 풀었을 때 진입변수를 선택할 수 없는 비가능이라는 의미이기 때문에 분지끝이 가능하다.

목적함수 값이 개선되지 않은 경우

패널티는 1회 선회연산을 실시했을 때의 목적함수값의 감소량이며, 다음의 경우 문제를 분지끝 시킬 수 있다.

$$z^t \ge (\text{부모문제의 목적함수값}) - \min(P_U, P_D) \tag{5.1}$$

분지변수를 선택하는 방법으로 패널티를 사용하여 분지변수로 x_i를 선택한다. 이 문제에 제약식 $x_i \le \lfloor \overline{b}_i \rfloor$과 $x_i \ge \lceil \overline{b}_i \rceil$를 추가하여 문제를 분지시켰을 때 이 문

제의 목적함수 감소량은 다음값보다 커지게 된다.

(부모 문제의 목적함수값)- $min(P_U, P_D)$ (5.2)

따라서 식 (5.1)과 같은 경우에는 현 상황에서 분지끝 판정을 내릴 수 있고, 식(5.1)의 경우 분지끝이 가능하다.

6. 부문제의 저장

분지한계법에서 분지과정을 통해 생성되는 부문제들의 수는 급격히 증가한기 때문에 효과적 관리가 중요하다.

생성되어지는 부문제들의 저장 - 이진나무(Binary Tree)의 이용

일반적으로 분지한계법에서는 하나의 부모문제에서 분지되어 두 개의 자식문제가 생성이 되어진다. 효율적 관리를 위한 자료구조로는 이진나무를 이용하며, 이것을 사용 부모문제와 자식문제 사이의 관계를 쉽게 알 수 있다.

기저정보의 저장

각 부문제의 최적기저를 보관하면 깊이우선탐색방법 뿐만 아니라 다른 탐색전략에서도 부모문제의 최적기저를 이용할 수 있으며, 부모문제의 최적기저를 초기해로 이용하여 쌍대단체법을 적용할 수 있다.

변환된 기저정보의 저장

부모문제의 기저 정보를 초기 기저로 잡는데 이용하는 것이 유리하다. 또, 부모문제와 자식문제 사이의 변화된 기저의 정보만 탐색나무에 보관하면 저장공간을 효율적으로 사용할 수 있다.

7. 원단체법과 쌍대단체법의 이용

MIPBB는 원문제는 원단체법을, 부문제들의 경우 부모문제의 최적기저를 이용 쌍대단체법을 적용한다.

<　2> 실험문제

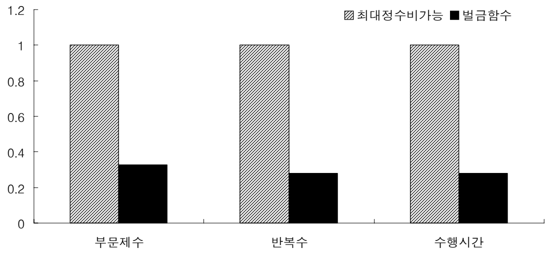
문제이름	ROWS	COLS	INT	0/1	문제이름	ROWS	COLS	INT	0/1
air01	23	771	771	ALL	misc02	39	59	58	ALL
air02	50	6774	6774	ALL	misc03	96	160	159	ALL
air03	124	10757	10757	ALL	misc04	1725	4897	30	ALL
air06	825	8627	8627	ALL	misc05	300	136	74	ALL
bell3a	123	133	71	39	misc06	820	1808	112	ALL
bell3b	123	133	71	39	mod008	6	319	319	ALL
bell4	105	117	64	34	mod013	62	96	48	ALL
bm23	20	27	27	ALL	p0033	16	33	33	ALL
cracpb1	143	572	572	ALL	p0040	23	40	40	ALL
dcmulti	290	548	75	ALL	p0201	133	201	201	ALL
diamond	4	2	2	ALL	p0282	241	282	282	ALL
dsbmip	1182	1886	192	160	p0291	252	291	291	ALL
egout	98	141	55	ALL	pipex	25	48	48	ALL
flugpl	18	18	11	0	rgn	24	180	100	ALL
gen	780	870	150	144	sample2	45	67	21	ALL
khb05250	101	1350	24	ALL	sentoy	30	60	60	ALL
lp4l	85	1086	1086	ALL	stein15	36	15	15	ALL
lseu	28	89	89	ALL	stein27	118	27	27	ALL
misc01	54	83	82	ALL	stein9	13	9	9	ALL

8. 실험결과

MIPBB는 C언어를 사용하여 작성된 분지한계법 프로그램으로서, 여기에 구현된 각각의 기법들의 효과들에 대해 실험적으로 분석하고자 한다. 실험대상문제는 <표 2>에 나타낸 MIPLIB이며[12], 실험환경은 Intel PentiumIII 600Mhz, RAM 128M이고 운영체제는 Redhat Linux 7.0이다. 실험결과를 정리하여 보면 다음과 같다.

분지전략

MIPBB에서는 여러 가지 분지전략들 중에서 최대비가능과 패널티를 이용한 분지전략을 구현하였다. 패널티를 이용한 분지전략이 최대비가능을 이용한 분지전략에 비해 생성되어지는 부문제의 수, LP-반복수, 소요되는 시간 등에서 모두 우수함을 알 수 있다.

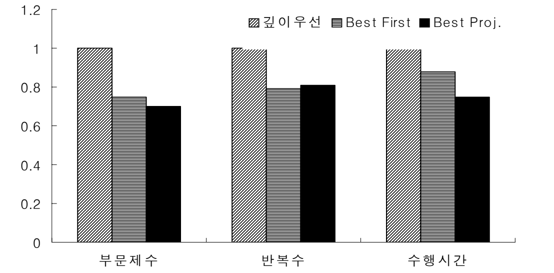


[.림 2] 분지전략 비교

탐색전략

① 탐색전략의 비교

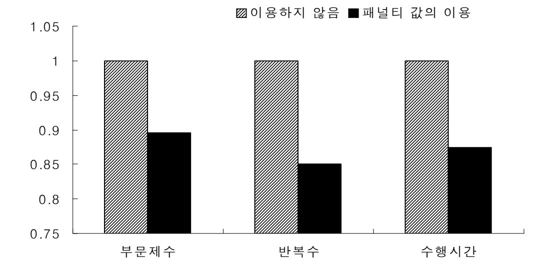
[그림 3]은 탐색전략에 부문제의 수, LP의 반복수, 총 수행시간을 실험한 결과이다. 부문제 수 측면에서는 Best-Projection과 Best-First탐색방법이 깊이우선탐색방법에 비해 각각 30%와 25% 감소 효과가 있다. LP-반복수 측면에서는 Best-First와 Best-Projection탐색방법이 각각 22%와 20%정도 감소효과가 있었다. 그리고 총 수행시간면에서는 Best-First와 Best-Projection탐색방법이 깊이우선탐색 방법에 비해 12%와 26%의 향상효과가 있다.



[.림 3]　　　　　1

② 탐색전략 - 패널티 값 이용의 효과

부문제의 목적함수 값의 추정치와 부모문제의 LP이완 목적함수 값을 사용하는 경우에 대해 비교해보기로 한다.



[.림 4]　　　　　2

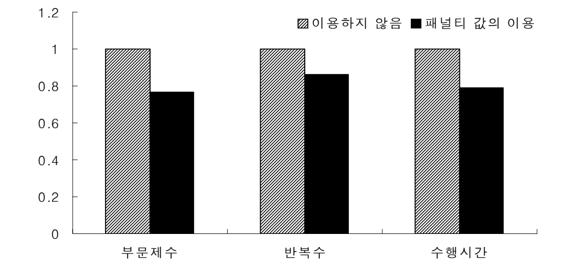
한계전략

MIPBB에서는 부문제가 비가능인 경우, 부문제의 LP이완 목적함수 값이 현재의 정수해보다 작은 경우에 대해 분지끝을 시킨다.

변환된 기저 정보의 저장

부모문제의 최적기저를 보관하고 있으면 깊이우선 탐색방법 이외의 방법에 대해서도 부모문제의 최적해에서 쌍대단체법을 이용할 수 있다. 이때, 변환된 기저만 보관

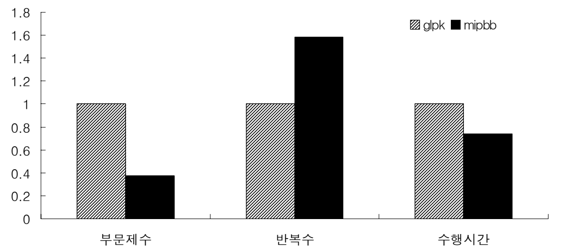
하면 기저와 관련된 정보를 저장하는 데 80%정도의 저장공간의 절감효과가 있다.



[.림 5] 한계전략의 비교

쌍대단체법과 원단체법의 사용

부문제를 쌍대단체법으로 푸는 경우 수치적 불안정성으로 인해 기저역행렬을 다시 계산해주는 과정이 발생하고, 프로그램의 수행시간의 성능 저하를 가져온다. 문제 중 Air02와 LP4L은 원단체법과 쌍대단체법을 전환하여 푼 결과 각각 80%와 90%의 수행속도 향상을 가져왔다.



[.림 6] GLPK-MIPBB 비교

9. 결론

본 연구에서는 혼합정수계획법의 해법인 분지한계법의 효과적 구현을 위해 분지, 탐색, 한계 전략, 부문제를 효율적 저장을 위한 여러 기법들의 구현 및 실험결과 분석을 하였다. 또한 부문제의 예측치의 한계 강화를 위해 분지변수 선택 시 계산되는 패널티를 이용하는 방법과 수치적으로 불안정한 부문제는 쌍대단체법에서 원단체법으로 전환하는 방법도 제시하였다. 분지한계법 프로그램 MIPBB는 공개용 우수 프로그램인 GLPK와 대등한 성능을 보이고 있다. 부문제수와 수행시간 면에서는 GLPK에 비해 우수한 성능을 보이지만 LP반복수 면에서는 GLPK에 비해 50-60%정도 많아 추후 연구가 필요하다.

참고문헌

[1] 박순달, 경영과학, 제4판, 민영사, 2000.
[2] 도승용, 이상욱, 임성묵, 박순달「혼합정수계획법을 위한 분지절단법의 효율적인 구현」, 춘계 경영과학/산업공학회 학술대회 CD, 2002, 5
[3] Cercile, Cordier, Hugues Marchand, Richard Laundy, Laurence A. Wolesy, "bc-opt : a branch-and-cut code for mixed integer programs", Math. Prog. Ser. A 86(1999), 335-353
[4] G. L. Nemhauser and M. W. P. Savelsbergh, and G. C. Sigismondi. MINTO, a Mixed INTeger Optimizer, Operations Research Letters, 15(1994), 47-58
[6] http ://www.cplex.com
[7] Stefan Thienel. ABACUS A Branch - and-Cut

< 3> (MIPBB-GLPK)

문제 이름	MIPBB			GLPK		
	부문 제수	LP 반복수	수행 시간	부문 제수	LP 반복수	수행 시간
air01	3	29	0.04	3	3	0.1
air02	16	4118	10.91	45	239	14.7
air03	2	318	4.51	3	6	22.8
air06	8	12152	145.71	35	2081	549.9
bell3a	48792	317193	656.28	53923	176931	189.7
bell3b	8705	25822	39.06	33475	38386	33.6
bell4	15174	38692	89.53			
bm23	337	2275	0.48	1257	2498	0.6
cracpb1	41	1237	1.53	10417	104684	223.9
dcmulti	1301	6628	18.9	7283	25533	53.8
diamond	1	6	0.01	7	4	0.01
dsbmip	52	5535	13.35	125	1209	23.2
egout	2725	5763	5.65	4773	4434	3.9
flugpl	1226	2094	0.59	3717	2744	0.6
gen	636	10435	21.66	933	7880	33.1
khb05250	2084	11145	24.68	5425	16852	41.4
lp4l	21	1054	0.9	183	1414	4
lseu	35410	323775	80.42	59595	107721	70.4
misc01	502	5890	4.45	643	3910	1.7
misc02	35	375	0.13	111	503	0.2
misc03	534	15020	28.52	1155	10338	9
misc04	15	2182	6.32	13	94	34.5
misc05	370	8238	9.59	621	4116	7.9
misc06	169	1565	6.22	117	352	5.4
mod008	10651	326671	80.92	182813	181289	523.1
mod013	374	2791	0.67	553	1441	0.5
p0033	342	1320	0.23	991	1339	0.2
p0040	35	184	0.02	83	127	0.01
p0201	716	14406	12.13	1515	7846	10.8
p0282	466	3057	3.85	1345	3231	4.7
p0291	67	316	0.5	227	440	0.6
pipex	1002	5617	1.04	3575	6309	1.4
rgn	3406	40924	8.39	3881	12274	4.5
sample2	200	701	0.22	305	663	0.2
sentoy	5097	228	14.15			
stein15	112	400	0.12	221	447	0.1
stein27	4624	17154	14.54	8035	19748	11.5
stein9	20	68	0.02	35	51	0.01

System, PhD thesis, Universitat zu koln, 1995
[8] Alexander Martin, General Mixed Integer Programming : Computational Issues for Branch-and-Cut Algorithm, Technical Report.
[9] Hanif. Sherali, Patrick J. Driscoll, Evolution and state-of-the-art in integer programming, Journal of Computational and Applied Mathematics, 124(2000) 319-340
[10] J. A. Tomlin. "An improved branch-and-bound method for integer programming method for integer programming", Operations Research, 19, pp1070-1075, 1971
[11] F. J. Dakin. "A tree search algorithm for mixed programming problems. Computer Journal, 8(3) pp 250-255, 1965
[12] R.E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsbergh. "An updated mixed integer programming library: MIPLIB 3.0. SIAM News.

[13] http://www.gnu.org/software/glpk/glpk.html
[14] MOSEK-2.5.1: http://www.mosek.com
[15] XPRESS-MP-13.26: http://www.dashoptimization.com/