

УДК 519.8

Н.Н. Писарук

## БИБЛИОТЕКА MIPCL ДЛЯ РЕШЕНИЯ ЗАДАЧ СМЕШАННО-ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ

*Рассматривается библиотека MIPCL – свободное программное обеспечение, предназначенное для быстрой и эффективной компьютерной реализации моделей смешанно целочисленного программирования (СЦП). Вычислительные эксперименты доказали, что сегодня MIPCL является наиболее эффективным свободным инструментом для решения задач СЦП. Библиотека MIPCL, документация и многочисленные примеры предоставляются на условиях лицензии GLPL (GNU Lesser Public License), поэтому MIPCL свободно доступна как для некоммерческого, так и коммерческого использования.*

### Введение

Задача СЦП представляет собой оптимизационную задачу с линейной целевой функцией и линейными ограничениями, в которой некоторые переменные должны принимать целочисленные значения. Многочисленные практические задачи из совершенно разных областей формулируются как задачи СЦП. Более обстоятельно о моделях и методах СЦП можно узнать, например, из источников [1–3].

В течение десятилетий после возникновения СЦП как раздела оптимизационной теории считалось, что СЦП является мощным инструментом для моделирования и почти бесполезно на практике, поскольку имеющиеся тогда компьютеры и программное обеспечение не могли решать практические задачи. За последние 20 лет эти устоявшиеся представления кардинально изменились: произошло впечатляющее улучшение качества программного обеспечения – как коммерческого, так и некоммерческого – для решения задач СЦП. Сегодня, используя более производительные компьютеры и лучшие программные продукты, можно решать многие классы трудных практических задач СЦП.

Некоммерческие программы для решения задач СЦП пока не могут конкурировать в скорости и надежности с лучшими коммерческими программами (CPLEX [4], GUROBY [5], Xpress-MP [6]), но они являются важной альтернативой дорогостоящему коммерческому программному обеспечению. В то же время для практических приложений умеренной размерности (а таких большинство) лучшие некоммерческие программы (SCIP [7], CBC [8], MIPCL [9]) почти так же эффективны, как и упомянутые выше коммерческие программы.

### 1. Библиотека MIPCL

MIPCL (Mixed Integer Programming Class Library) – это библиотека для решения задач СЦП. Она полностью реализована на C++ и может использоваться самостоятельно как разделяемая (динамическая) библиотека в C++ проектах, вместе с моделирующей оболочкой MIPshell, как модуль в Python программах или как утилита (самостоятельная программа) для решения задач СЦП, записанных в текстовых файлах в MPS-формате.

Подробное описание всех функций MIPCL можно найти в [9].

MIPCL реализована как семейство C++ классов. Два главных класса – это SMIP и CLP. SMIP есть программная система для решения задач СЦП, а CLP – система для решения задач линейного программирования (ЛП).

MIPCL поддерживает следующие функции:

- реализацию алгоритмов ветвей и сечений и алгоритмов ветвей и оценивания [2, 3];
- автоматическую переформулировку решаемых задач с целью их упрощения;
- устранение симметрии;

– реализацию прямых эвристик, которые пытаются «округлять» решения релаксационных задач;

– генерирование отсечений и столбцов;

– «сильное» ветвление.

MIPCL API позволяет разработчикам с легкостью интегрировать оптимизацию в свои приложения. Интерфейсы для разработчиков удобны и просты в использовании. MIPCL – очень гибкая программная система, что обусловлено большим количеством виртуальных функций, которые позволяют разработчикам переопределять (перепрограммировать) многие ее компоненты, чтобы лучше учесть специфику решаемой задачи. Такими переопределяемыми компонентами являются:

– процедуры отделения, в которых должны строиться специфичные для решаемой задачи отсечения;

– процедуры для генерирования столбцов в алгоритмах ветвей и оценивания;

– прямые эвристики для округления решений релаксационных задач;

– селекторы узлов, которые направляют обход узлов дерева поиска;

– процедуры ветвления, которые задают правила для разбиения задач на подзадачи;

– методы для переформулировки решаемых задач с целью их упрощения;

– процедуры и структуры данных для хранения отсечений и генерируемых столбцов.

MIPCL распространяется с понятной и обстоятельной документацией, а также множеством примеров, которые доступны в исходных кодах и должны помочь разработчикам программного обеспечения быстро освоить приемы работы с библиотекой.

### 1.1. Мультипоточность

Современные многоядерные персональные компьютеры обеспечивают аппаратную и программную среду для эффективных параллельных вычислений. Поэтому дизайны большинства современных библиотек для решения задач СЦП (включая MIPCL) были переработаны с целью распараллелить работу алгоритмов ветвей и сечений и алгоритмов ветвей и оценивания.

Разработка и отладка сложных мультипоточных приложений – далеко не тривиальная задача. При разработке мультипоточной библиотеки MIPCL главной задачей было не добавить будущим пользователям дополнительных сложностей. В результате, если пользователь в своем приложении не перегружает никаких MIPCL-функций, то беспокоиться ему не о чем: приложение будет работать одинаково хорошо как с однопоточной, так и с многопоточной библиотеками. Однако если приложение генерирует отсечения или столбцы, реализует более подходящую для решаемой задачи стратегию ветвления или просто переопределяет любую виртуальную функцию базового класса SCIP, то в своем производном классе пользователю необходимо реализовать конструктор клонирования вместе с интерфейсом, который MIPCL использует для создания объектов производного класса. Конструктор клонирования вызывается при создании нового потока. Он очень похож на конструктор копирования за исключением того, что не должен выделять память для хранения разделяемых (общих для всех потоков) данных.

### 1.2. Эффективность

Для тестирования MIPCL утилита *mps\_mipcl* с лимитом времени 2 ч на тест решала задачи СЦП из библиотеки тестов MIPLIB2010 [10] (основное семейство тестов). Результаты тестов фиксировались с помощью MIPLIB2010-скриптов. Характеристики компьютера, на котором проводились тесты, следующие: AMD FX8120, 8 ядер, 8 GB, 3,1 GHz. В сжатой форме эти результаты представлены в последнем столбце таблицы.

Результаты тестов

	CBC	CPLEX	SCIP	SCIPS	GUROBI	XPRESS	MIPCL
Решены	61	86	72	67	86	86	82
Прерваны	24	1	10	17	1	1	5
Не решены	2	0	5	3	0	0	0
GMRT	17,4	1,05	8,19	12,7	1	1,3	4,1

Для сравнения MIPCL с лучшими конкурирующими программами (коммерческими и некоммерческими) были использованы результаты тестов, проведенных Х. Миттельманом (<http://mittelmann@asu.edu>). В этих тестах с лимитом времени в 2 ч решались задачи из той же библиотеки MIPLIB2010 с помощью следующих программ:

CPLEX: CPLEX-12.6.1 [4];

GUROBY: GUROBY-6.0.0 [5];

SCIP: ug[SCIP/cpx]-3.1.1 (разрабатываемая параллельная версия SCIP+CPLEX) (<http://www.zib.de>);

SCIPS: ug[SCIP/spx]-3.1.1 (разрабатываемая параллельная версия SCIP+SOPLEX) (<http://www.zib.de>);

CBC: CBC-2.9.4 [11];

XPRESS: XPRESS-7.8.0 [6].

Отметим, что среди этих программ некоммерческими являются только SCIPS и CBC.

Результаты MIPCL сравнивались с результатами конкурентов, полученными на компьютере со следующими характеристиками: Intel i7-2600, 4 ядра, 16 GB, 3,4 GHz. Заметим, что производительность такого компьютера даже несколько выше производительности компьютера, на котором осуществлялось тестирование MIPCL.

В компактном виде результаты всех тестов представлены в таблице, где GMRT означает отношение геометрического среднего времен решения всех тестов к такому же показателю самой быстрой программы.

## 2. Моделирующая оболочка MIPshell

Каждый, кто когда-либо пробовал применять математическое программирование на практике, знает, что в большинстве случаев это непростое упражнение. Для упрощения моделирования и решения оптимизационных задач были разработаны языки оптимизационного моделирования OPL [12], AMPL [13] и др. MIPshell не является таким языком. Это просто программная оболочка для облегчения моделирования и решения задач ЛП и СЦП с использованием библиотеки MIPCL. MIPshell включает в себя:

- семейство C++ классов, разработанных с использованием библиотек STL, для представления переменных, ограничений, множеств, векторов, ассоциативных массивов (словарей) и других структур данных;

- набор функций для упрощения записи оптимизационных задач;

- препроцессор, который транслирует новые (не из C++) операторы моделирования в фрагменты кода на языке C++.

Существует несколько причин, по которым оболочка MIPshell не трансформировалась в полноценный язык оптимизационного моделирования. При разработке сложных практических приложений требуется программная среда для реализации графических интерфейсов пользователя, интерфейсов для работы с базами данных, сетевыми протоколами (для связи с удаленными серверами) и еще много чего другого. Ни один из существующих языков оптимизационного моделирования не реализует даже минимум из необходимого. Между тем современные средства разработки программного обеспечения, такие как Qt или Microsoft Visual Studio, имеют в своем составе все необходимое. Разработчики могут легко интегрировать MIPCL вместе с MIPshell в любую из таких систем разработки.

**Демонстрационный пример.** Рассмотрим следующую задачу СЦП:

$$\sum_{e \in E} (f_e y_e + c_e x_e) \rightarrow \min;$$

$$\sum_{e \in E: t_e = v} x_e - \sum_{e \in E: h_e = v} x_e = d_v, v \in V,$$

$$0 \leq x_e \leq u_e y_e, e \in E,$$

$$y_e \in \{0, 1\}, e \in E,$$

которая является формулировкой сетевой транспортной задачи с постоянными издержками (см. [2]). Здесь  $G=(V, E)$  – ориентированный граф,  $d_v$  – спрос в узле (вершине)  $v \in V$  на некоторый продукт, а каждой дуге  $e=(v, w) \in E$ , соединяющей начальную вершину  $t_e=v$  с конечной вершиной  $h_e=w$ , приписаны пропускная способность  $u_e$ , а также постоянные издержки  $f_e$  и переменные издержки  $c_e$ :  $f_e + c_e x_e$  есть стоимость транспортировки  $x_e > 0$  единиц продукта по дуге  $e$  (если  $x_e=0$ , то ничего платить не нужно).

В MIPshell задача СЦП записывается следующим образом:

```
int e;
VAR_VECTOR x("x",REAL_GE,E);
VAR_VECTOR y("y",BIN,E);

minimize(sum(e in E) (f(e)*y(e) + c(e)*x(e)));
forall(v in V)
    sum(e in E: h(e)==v) x(e) - sum(e in E: t(e)==v) x(e) == d(v);
forall(e in E)
    x(e) <= u(e)*y(e);
```

Чтобы облегчить изучение оболочки, MIPshell поставляется с ясной и обстоятельной документацией, а также большим количеством примеров. Поскольку все примеры доступны в исходных кодах, они будут полезны и при изучении СЦП.

### 3. Python-MIPCL

Python-MIPCL представляет собой модуль на языке программирования Python. По своему назначению и функциональности он аналогичен оболочке MIPshell. Python-MIPCL включает в себя следующие компоненты:

- разделяемую библиотеку mipcl.so (на windows компьютерах mipcl.pyd), которая является библиотекой MIPCL с дополнительным интерфейсом для вызовов из программ на языке Python;

- набор классов на языке Python, которые спроектированы для представления переменных, ограничений и оптимизационных задач;

- семейства функций для упрощения записи формулировок задач СЦП.

Python-MIPCL реализует только часть функциональности базовой оболочки MIPshell. В частности, Python-MIPCL не позволяет реализовывать приложения, которые должны генерировать отсечения. Причина заключается в том, что реализовывать такие приложения на скриптовом языке неэффективно, а подавляющее большинство практических приложений этого и не требует. Далеко не каждый, кто использует оптимизацию на практике, знает C++. Python прост для начинающих и поэтому Python-MIPCL поможет расширить круг пользователей MIPCL.

Для сравнения функциональности Python-MIPCL с MIPshell ниже представлен фрагмент кода на языке Python, в котором записана модель рассмотренной задачи СЦП:

```
class Fcnf(Problem):
    def model(self,t,h,u,d,f,c):
        m = len(t)
        n = len(d)
        x = VarVector([m],"x",REAL|GE)
        y = VarVector([m],"y",BIN)

        minimize(sum_(f[e]*y[e] + c[e]*x[e] for e in range(0,m)))
        for v in range(0,n):
            sum_(x[e] for e in range(0,m) if h[e]==v) -\
```

```

sum_(x[e] for e in range(0,m) if t[e]==v) == d[v]
for e in range(0,m):
    x[e] <= u[e]*y[e]

```

### Список литературы

1. 50 Years of Integer Programming 1958–2008 / M. Jürgen [et al.]. – Berlin : Springer, 2009.
2. Wolsey, L.A. Integer Programming / L.A. Wolsey. – Wiley, 1998. – 288 p.
3. Pисарук, N.N. Models and methods of mixed-integer programming / N.N. Pисарук. – Minsk : Belarus State University, 2010.
4. IBM. IBM ILOG CPLEX Optimization Studio [Electronic resource]. – 2015. – Mode of access : <http://www-01.ibm.com/software/integration/optimisation/cplex-optimization-studio>. – Date of access : 06.03.2015.
5. Guroby Optimization Inc. Guroby Optimizer [Electronic resource]. – 2012. – Mode of access : <http://www.guroby.com/welcome.html>. – Date of access : 06.03.2015.
6. Dash Optimization. Xpress-MP [Electronic resource]. – 2015. – Mode of access : <http://www.dashoptimization.com>. – Date of access : 06.03.2015.
7. Achterberg, T. SCIP: solving constraint integer programs / T. Achterberg // Math. Prog. Comp. – 2009. – Vol. 1. – P. 1–41.
8. COIN-OR. Computational Infrastructure for Operations Research [Electronic resource]. – 2015. – Mode of access : <http://www.coin-or.org>. – Date of access : 06.03.2015.
9. MIPCL Reference Manual [Electronic resource]. – 2015. – Mode of access : <http://mipcl-cpp.appspot.com/static/docs/mipcl/html/index.html>. – Date of access : 06.03.2015.
10. MIPLIB2010 [Electronic resource]. – 2015. – Mode of access : <http://miplib.zib.de>. – Date of access : 06.03.2015.
11. Forrest, J.J.H. COIN branch and cut / J.J.H. Forrest [Electronic resource]. – 2015. – Mode of access : <http://www.coin-or.org>. – Date of access : 06.03.2015.
12. Modeling with OPL [Electronic resource]. – 2015. – Mode of access : <http://www-01.ibm.com/software/commerce/optimization/modeling/index.html>. – Date of access : 06.03.2015.
13. Fourer, R. AMPL: A Modeling Language for Mathematical Programming / R. Fourer, B.W. Kernigan. – Duxbury Press, 2002.

Поступила 20.05.2016

*Белорусский государственный университет,  
Минск, ул. К. Маркса, 31  
e-mail: pizaruk@yandex.by*

**N.N. Pисарук**

### MIPCL LIBRARY FOR SOLVING MIXED INTEGER PROGRAMMING PROBLEMS

The Mixed Integer Programming Class Library (MIPCL) is a free software designed for implementing mixed integer programming models quickly, easily, and efficiently. Computational experiments show that currently MIPCL is one of the best noncommercial mixed-integer programming solvers. MIPCL libraries, documentation and examples are provided under the terms of the GNU Lesser Public License. Therefore, MIPCL is equally freely available for noncommercial and commercial use.