Combinatorial Benders' Cuts

Gianni Codato
DEI, University of Padova, Italy

Matteo Fischetti
DEI, University of Padova, Italy
matteo.fischetti@unipd.it

IPCO X, New York, June 2004

Introduction

- Consider a 0-1 ILP of the form

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

Introduction

- Consider a 0-1 ILP of the form

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by a set of **"conditional" linear constraints** involving additional continuous variables y

Introduction

- Consider a 0-1 ILP of the form

$$\min\{c^T x : Fx \le g, x \in \{0,1\}^n\}$$

amended by a set of **"conditional" linear constraints** involving additional continuous variables y

$$x_{j(i)} = 1 \Rightarrow a_i^T y \ge b_i \quad \text{for all } i \in I$$

Introduction

- Consider a 0-1 ILP of the form

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by a set of **"conditional" linear constraints** involving additional continuous variables y

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i \quad \text{for all } i \in I$$

plus a (possibly empty) set of **"unconditional" linear constraints** on the continuous variables y

$$Dy \geq e$$

Introduction

- Consider a 0-1 ILP of the form

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by a set of **"conditional" linear constraints** involving additional continuous variables y

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i \quad \text{for all } i \in I$$

plus a (possibly empty) set of **"unconditional" linear constraints** on the continuous variables y

$$Dy \geq e$$

- Note: the continuous variables y do not appear in the objective function—they are only introduced to force some feasibility properties of the x's.

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

- **Map Labelling Problem**: placing as many rectangular labels as possible (without overlap) in a given 2-dimensional map

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

- **Map Labelling Problem**: placing as many rectangular labels as possible (without overlap) in a given 2-dimensional map
  - binary variables are associated to the relative position of the pairs of labels to be placed

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

- **Map Labelling Problem**: placing as many rectangular labels as possible (without overlap) in a given 2-dimensional map
  - binary variables are associated to the relative position of the pairs of labels to be placed
  - continuous variables give the placement coordinates of the labels

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

- **Map Labelling Problem**: placing as many rectangular labels as possible (without overlap) in a given 2-dimensional map
  - binary variables are associated to the relative position of the pairs of labels to be placed
  - continuous variables give the placement coordinates of the labels
  - **conditional** constraints are of the type "if label i is placed on the right of label j, then the horizontal coordinates of i and j must obey a certain linear inequality giving a suitable separation condition"

Examples

- **Asymmetric Travelling Salesman Problem with Time Windows**
  - binary variables x_{ij} = usual arc variables
  - continuous variables y_i = arrival time at city i
  - **conditional** constraints are the logical implications:

$$x_{ij} = 1 \Rightarrow y_j \geq y_i + travel\_time(i, j)$$

  - **unconditional** constraints limit the arrival time at each city i:

$$early\_arrival\_time(i) \leq y_i \leq late\_arrival\_time(i).$$

- **Map Labelling Problem**: placing as many rectangular labels as possible (without overlap) in a given 2-dimensional map
  - binary variables are associated to the relative position of the pairs of labels to be placed
  - continuous variables give the placement coordinates of the labels
  - **conditional** constraints are of the type "if label i is placed on the right of label j, then the horizontal coordinates of i and j must obey a certain linear inequality giving a suitable separation condition"
  - **unconditional** constraints limit the label coordinates

The (in)famous big-M method

- Conditional constraints

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i \quad \text{for all } i \in I$$

typically modeled as follows (for sufficiently large $M_i > 0$):

$$a_i^T y \geq b_i - M_i(1 - x_{j(i)}) \quad \text{for all } i \in I$$

The (in)famous big-M method

- Conditional constraints

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i \quad \text{for all } i \in I$$

typically modeled as follows (for sufficiently large $M_i > 0$):

$$a_i^T y \geq b_i - M_i(1 - x_{j(i)}) \quad \text{for all } i \in I$$

- **Drawbacks**:
  - **Very poor** LP relaxation
  - **Large mixed-integer model** involving both x and y variables

The MIP solver is "carrying on its shoulders" the burden of *all* additional constraints and variables at *all* branch-decision nodes, while these become relevant only when the corresponding $x_{j(i)}$ attain value 1 (typically, because of branching).

The (in)famous big-M method

- Conditional constraints

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i \quad \text{for all } i \in I$$

typically modeled as follows (for sufficiently large $M_i > 0$):

$$a_i^T y \geq b_i - M_i(1 - x_{j(i)}) \quad \text{for all } i \in I$$

- **Drawbacks**:
  - **Very poor** LP relaxation
  - **Large mixed-integer model** involving both x and y variables

  The MIP solver is "carrying on its shoulders" the burden of *all* additional constraints and variables at *all* branch-decision nodes, while these become relevant only when the corresponding $x_{j(i)}$ attain value 1 (typically, because of branching).

- Note: one can get rid of the y variables by using **Benders' decomposition**, but this just a way to speed-up the LP solution—the resulting cuts are weak and still depend on the big-M values.

Combinatorial Benders' cuts

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i, \text{ for all } i \in I$$

$$Dy \geq e$$

Combinatorial Benders' cuts

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i, \text{ for all } i \in I$$

$$Dy \geq e$$

- We work on the space of the x-variables only, as in the classical Benders's approach, **but** ...

Combinatorial Benders' cuts

$$x_{j(i)} = 1 \Rightarrow a_i^T y \geq b_i, \text{ for all } i \in I$$

$$Dy \geq e$$

- We work on the space of the x-variables only, as in the classical Benders's approach, **but** ...
- ... we model the constraints involving the y variables through the following **Combinatorial Benders' (CB) cuts**:

$$\sum_{i \in C} x_{j(i)} \leq |C| - 1$$

where $C \subseteq I$ is an **inclusion-minimal** set such that the linear system

$$SLAVE(C) := \begin{cases} a_i^T y \geq b_i, \text{ for all } i \in C \\ Dy \geq e \end{cases}$$

has no feasible (continuous) solution y.

CB cut separation

- CB cut violated by a given $x^* \in [0,1]^n$ iff $\sum_{i \in C}(1 - x^*_{j(i)}) < 1$, hence;

  (i) weigh each conditional constraint $a_i^T y \leq b_i$ by $1 - x^*_{j(i)}$;

  (ii) weigh each unconditional constraint in $Dy \geq e$ by 0;

  (iii) look for a minimum-weight MIS (minimal infeasible system, or IIS) of the resulting weighted system (NP-hard).

CB cut separation

- CB cut violated by a given $x^* \in [0,1]^n$ iff $\sum_{i \in C}(1 - x^*_{j(i)}) < 1$, hence;

  (i) weigh each conditional constraint $a_i^T y \le b_i$ by $1 - x^*_{j(i)}$;

  (ii) weigh each unconditional constraint in $Dy \ge e$ by 0;

  (iii) look for a minimum-weight MIS (minimal infeasible system, or IIS) of the resulting weighted system (NP-hard).

- A simple polynomial-time heuristic:

  - start with $C := \{i \in I : x^*_{j(i)} = 1\}$,
  - verify the infeasibility of

$$SLAVE(C) := \begin{cases} a_i^T y \ge b_i, \text{ for all } i \in C \\ Dy \ge e \end{cases}$$

    by classical LP tools,
  - make C inclusion-minimal in a greedy way.

CB cut separation

- CB cut violated by a given $x^* \in [0,1]^n$ iff $\sum_{i \in C} (1 - x^*_{j(i)}) < 1$, hence;

  (i) weigh each conditional constraint $a_i^T y \leq b_i$ by $1 - x^*_{j(i)}$;

  (ii) weigh each unconditional constraint in $Dy \geq e$ by 0;

  (iii) look for a minimum-weight MIS (minimal infeasible system, or IIS) of the resulting weighted system (NP-hard).

- A simple polynomial-time heuristic:

  - start with $C := \{i \in I : x^*_{j(i)} = 1\}$,
  - verify the infeasibility of

$$SLAVE(C) := \begin{cases} a_i^T y \geq b_i, \text{ for all } i \in C \\ Dy \geq e \end{cases}$$

    by classical LP tools,
  - make C inclusion-minimal in a greedy way.

- The above heuristic is indeed **exact** when x^* is integer.

A Branch&Cut solution scheme

- Work in the x space. At each branching node:
    1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by the CB cuts generated so far

A Branch&Cut solution scheme

- Work in the x space. At each branching node:

  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

  amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

A Branch&Cut solution scheme

- Work in the x space. At each branching node:
  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

  amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

  Notes:

A Branch&Cut solution scheme

- Work in the x space. At each branching node:

  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

  amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

  Notes:

- The generated CB cuts automatically **distill** combinatorial information hidden in the MIP model, triggering the activation of other classes of combinatorial cuts, including Caprara-Fischetti $\{0, 1/2\}$-cuts.

A Branch&Cut solution scheme

- Work in the x space. At each branching node:
  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

     amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

  Notes:

- The generated CB cuts automatically **distill** combinatorial information hidden in the MIP model, triggering the activation of other classes of combinatorial cuts, including Caprara-Fischetti $\{0, 1/2\}$-cuts.
- The role of the big-M terms in the MIP model vanishes–only logical implications are relevant, no matter the way they are modelled $\Rightarrow$ logical implications can be stated explicitly in the model.

A Branch&Cut solution scheme

- Work in the x space. At each branching node:
  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

Notes:

- The generated CB cuts automatically **distill** combinatorial information hidden in the MIP model, triggering the activation of other classes of combinatorial cuts, including Caprara-Fischetti $\{0, 1/2\}$-cuts.
- The role of the big-M terms in the MIP model vanishes–only logical implications are relevant, no matter the way they are modelled $\Rightarrow$ logical implications can be stated explicitly in the model.
- Related to the concept of **nogoods** (minimal infeasible configurations of the binary variables) used in Constraint Programming (Hooker and Ottosson, 2003, and Thorsteinsson, 2001).

A Branch&Cut solution scheme

- Work in the x space. At each branching node:
  1. solve the LP relaxation of the *Master Problem*

$$\min\{c^T x : Fx \leq g, x \in \{0,1\}^n\}$$

amended by the CB cuts generated so far

  2. apply the CB separation heuristic so as to generate new violated combinatorial cuts—and to assert the feasibility of x^*, if integer.

Notes:

- The generated CB cuts automatically **distill** combinatorial information hidden in the MIP model, triggering the activation of other classes of combinatorial cuts, including Caprara-Fischetti $\{0, 1/2\}$-cuts.
- The role of the big-M terms in the MIP model vanishes–only logical implications are relevant, no matter the way they are modelled $\Rightarrow$ logical implications can be stated explicitly in the model.
- Related to the concept of **nogoods** (minimal infeasible configurations of the binary variables) used in Constraint Programming (Hooker and Ottosson, 2003, and Thorsteinsson, 2001).
- Interesting connections with Chvátal's **resolution search** and with Glover-Tangedhal's **dynamic branch and bound**.

A more general framework

- Consider a MIP problem with the following structure:

$$z^* := \min \quad c^T x + 0^T y \tag{1}$$

$$\text{s.t.} \quad Fx \le g \tag{2}$$

$$Mx + Ay \ge b \tag{3}$$

$$Dy \ge e \tag{4}$$

$$x_j \in \{0,1\} \quad \text{for } j \in B \tag{5}$$

A more general framework

- Consider a MIP problem with the following structure:

$$
\begin{aligned}
z^* := \text{min} \quad & c^T x + 0^T y && (1)\\
\text{s.t.} \quad & Fx \le g && (2)\\
& Mx + Ay \ge b && (3)\\
& Dy \ge e && (4)\\
& x_j \in \{0,1\} \quad \text{for } j \in B && (5)
\end{aligned}
$$

- We assume linking constraints (3) are of the type:

$$m_{i,j(i)} x_{j(i)} + a_i^T y \ge b_i \text{ for all } i \in I \tag{6}$$

A more general framework

- Consider a MIP problem with the following structure:

$$z^* := \min \quad c^T x + 0^T y \tag{1}$$

$$\text{s.t.} \quad Fx \le g \tag{2}$$

$$Mx + Ay \ge b \tag{3}$$

$$Dy \ge e \tag{4}$$

$$x_j \in \{0, 1\} \quad \text{for } j \in B \tag{5}$$

- We assume linking constraints (3) are of the type:

$$m_{i,j(i)} x_{j(i)} + a_i^T y \ge b_i \text{ for all } i \in I \tag{6}$$

- A useful trick: introduce a continuous copy x_j^c of each binary variable x_j, $j \in B$, and link the two copies through the constraints:

$$x_j = x_j^c \quad \text{for } j \in B \tag{7}$$

Master-slave solution scheme

- *MASTER*:

$$z^* = \min \quad c^T x \tag{8}$$

$$\text{s.t.} \quad Fx \le g \quad \text{(plus additional cuts, if any)} \tag{9}$$

$$x_j \in \{0, 1\} \ \text{ for } \ j \in B \tag{10}$$

Master-slave solution scheme

- *MASTER*:

$$z^* = \min \quad c^T x \tag{8}$$

$$\text{s.t.} \quad Fx \le g \quad \text{(plus additional cuts, if any)} \tag{9}$$

$$x_j \in \{0,1\} \ \text{ for } \ j \in B \tag{10}$$

- *SLAVE*(x), a linear system parametrized by x:

$$Ay \ge b - Mx \tag{11}$$

$$Dy \ge e \tag{12}$$

Master-slave solution scheme

- *MASTER*:

$$z^* = \min \quad c^T x \tag{8}$$

$$\text{s.t.} \quad Fx \leq g \quad \text{(plus additional cuts, if any)} \tag{9}$$

$$x_j \in \{0, 1\} \quad \text{for} \quad j \in B \tag{10}$$

- *SLAVE(x)*, a linear system parametrized by x:

$$Ay \geq b - Mx \tag{11}$$

$$Dy \geq e \tag{12}$$

- If x^* integer and $SLAVE(x^*)$ infeasible, take any MIS of SLAVE(x^*) involving the rows of A indexed by C (say)

Master-slave solution scheme

- *MASTER*:

$$z^* = \min \quad c^T x \tag{8}$$

$$\text{s.t.} \quad Fx \leq g \quad \text{(plus additional cuts, if any)} \tag{9}$$

$$x_j \in \{0, 1\} \quad \text{for} \quad j \in B \tag{10}$$

- *SLAVE(x)*, a linear system parametrized by x:

$$Ay \geq b - Mx \tag{11}$$

$$Dy \geq e \tag{12}$$

- If x^* integer and $SLAVE(x^*)$ infeasible, take any MIS of SLAVE(x^*) involving the rows of A indexed by C (say) $\Rightarrow$ at least one binary variable $x_{j(i)}$, $i \in C$, has to be changed in order to break the infeasibility

Master-slave solution scheme

- *MASTER*:

$$z^* = \min \quad c^T x \tag{8}$$

$$\text{s.t.} \quad Fx \leq g \quad \text{(plus additional cuts, if any)} \tag{9}$$

$$x_j \in \{0,1\} \quad \text{for} \quad j \in B \tag{10}$$

- *SLAVE(x)*, a linear system parametrized by x:

$$Ay \geq b - Mx \tag{11}$$

$$Dy \geq e \tag{12}$$

- If x^* integer and $SLAVE(x^*)$ infeasible, take any MIS of SLAVE(x^*) involving the rows of A indexed by C (say) $\Rightarrow$ at least one binary variable $x_{j(i)}$, $i \in C$, has to be changed in order to break the infeasibility $\Rightarrow$ Combinatorial Benders' (CB) cut:

$$\sum_{i \in C : x^*_{j(i)} = 0} x_j + \sum_{i \in C : x^*_{j(i)} = 1} (1 - x_j) \geq 1. \tag{13}$$

Computational Results

- Implementation in C++ embedded within the ILOG-Cplex Concert Technology 1.2 framework (ILOG-Cplex 8.1 MIP solver).
- Experiments on a PC AMD Athlon 2100+ with 1 GByte RAM, with a time-limit of 3 CPU hours for each run.

Computational Results

- Implementation in C++ embedded within the ILOG-Cplex Concert Technology 1.2 framework (ILOG-Cplex 8.1 MIP solver).
- Experiments on a PC AMD Athlon 2100+ with 1 GByte RAM, with a time-limit of 3 CPU hours for each run.
- The test-bed:

  1. **Map Labelling** (max. problem): placing as many rectangular labels as possible (without overlap) in a given map (Mützel and Klau, 2003)

  2. **Two-Group Statistical Classification** (discriminant analysis): given a population whose members can be divided into two distinct classes, define a linear function of some available measures so as to decide whether a new member belongs to the first or second class (Rubin, 1997).

Computational Results

- Implementation in C++ embedded within the ILOG-Cplex Concert Technology 1.2 framework (ILOG-Cplex 8.1 MIP solver).

- Experiments on a PC AMD Athlon 2100+ with 1 GByte RAM, with a time-limit of 3 CPU hours for each run.

- The test-bed:

  1. **Map Labelling** (max. problem): placing as many rectangular labels as possible (without overlap) in a given map (Mützel and Klau, 2003)

  2. **Two-Group Statistical Classification** (discriminant analysis): given a population whose members can be divided into two distinct classes, define a linear function of some available measures so as to decide whether a new member belongs to the first or second class (Rubin, 1997).

- All instances have been processed twice:

  Cplex: the original MIP model is solved through the commercial ILOG-Cplex 8.1 solver (with default settings), and

  CBC: the master/slave reformulation is solved by using CB cuts and $\{0, 1/2\}$-cuts (still using the ILOG-Cplex 8.1 library).

Subset 1		**Cplex**	**CBC**	**Statistics**	
File name	**opt.**	**Time hh : mm : ss**	**Time hh : mm : ss**	**t.Cplex/ t.CBC**	**t.CBC/ t.Cplex**
MAP LABELLING					
CMS 600-1	600	1 : 08 : 41	0 : 04 : 34	15.0	6.6%
STAT. ANALYSIS					
Chorales-116	24	1 : 24 : 52	0 : 10 : 18	8.2	12.1%
Horse-colic-151	5	0 : 04 : 50	0 : 00 : 23	12.6	7.9%
Iris-150	18	0 : 09 : 29	0 : 01 : 10	8.1	12.3%
Credit-300	8	0 : 19 : 35	0 : 00 : 02	587.5	0.2%
Lymphography-142	5	0 : 00 : 11	0 : 00 : 01	11.0	9.1%
Mech-analysis-107	7	0 : 00 : 05	0 : 00 : 01	5.0	20.0%
Mech-analysis-137	18	0 : 07 : 44	0 : 00 : 27	17.2	5.8%
Monks-tr-122	13	0 : 02 : 05	0 : 00 : 05	25.0	4.0%
Pb-gr-txt-198	11	0 : 04 : 21	0 : 00 : 05	52.2	1.9%
Pb-pict-txt-444	7	0 : 02 : 07	0 : 00 : 02	63.5	1.6%
Pb-hl-pict-277	10	0 : 04 : 17	0 : 00 : 27	9.5	10.5%
Postoperative-88	16	0 : 15 : 16	0 : 00 : 01	916.0	0.1%
BV-OS-282	6	0 : 05 : 13	0 : 00 : 24	13.0	7.7%
Opel-Saab-80	6	0 : 01 : 03	0 : 00 : 13	4.8	20.6%
Bus-Van-437	6	0 : 09 : 17	0 : 00 : 28	19.9	5.0%
HouseVotes84-435	6	0 : 04 : 59	0 : 00 : 11	27.2	3.7%
Water-treat-206	4	0 : 01 : 10	0 : 00 : 06	11.7	8.6%
Water-treat-213	5	0 : 17 : 00	0 : 00 : 51	20.0	5.0%
TOTALS		**8 : 29 : 51**	**0 : 24 : 11**	**21.1**	**5%**

Table 1: Problems solved to proven optimality by both Cplex and CBC

File name (Subset 1)	**n. nodes Cplex**	**n. nodes CBC**
STAT. ANALYSIS		
Chorales-116	10,329,312	20,382
Horse-colic-151	135,018	2,184
Iris-150	970,659	1,290
Credit-300	176,956	66
Lymphography-142	8,157	106
Mech-analysis-107	11,101	68
Mech-analysis-137	938,088	1,888
Monks-tr-122	262,431	357
Pb-gr-txt-198	135,980	110
Pb-pict-txt-277	71,031	1,026
Pb-hl-pict-444	22,047	115
Postoperative-88	2,282,109	171
BV-OS-282	56,652	1,044
Opel-Saab-80	87,542	7,314
Bus-Van-437	55,224	6,795
HouseVotes84-435	42,928	734
Water-treat-206	12,860	482
Water-treat-213	168,656	4,036
MAP LABELLING		
CMS 600-1	110,138	14

Table 2: Number of branch-decision nodes enumerated by Cplex and by CBC, respectively

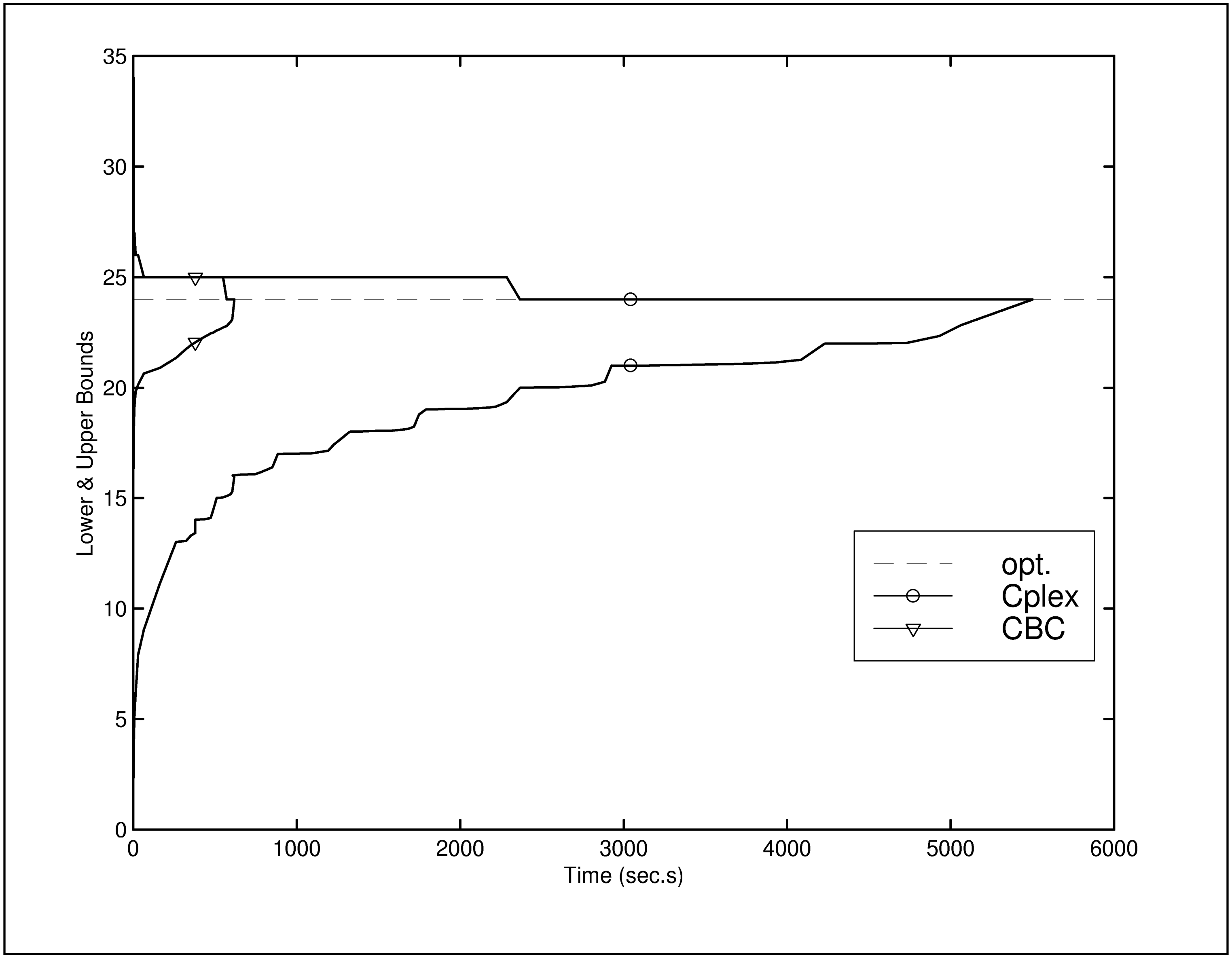

Figure 1: Optimizing the statistical-analysis instance Chorales-116 (minimization problem)

Subset 2	**Cplex**					**CBC**	
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem MB**	**Time hh:mm:ss**	**opt.**
Chorales-134	0: 36 :23	33	16.0	51%	371	0: 36 :23	30
	3: 00 :58	30	21.1	30%	992		
Chorales-107	0: 04 :19	28	12.1	57%	61	0: 04 :19	27
	3: 01 :27	27	22.2	18%	711		
Breast-Cancer-600	0: 00 :13	108	1.5	99%	9	0: 00 :13	16
	3: 00 :11	16	13.2	18%	45		
Bridges-132	0: 03 :39	33	5.1	85%	44	0: 03 :39	23
	3: 01 :09	23	10.0	56%	1406		
Mech-analysis-152	0: 34 :12	22	12.1	45%	328	0: 34 :12	21
	3: 00 :50	21	16.1	24%	865		
Monks-tr-124	0: 01 :55	27	8.1	70%	25	0: 01 :55	24
	3: 00 :35	24	20.0	17%	381		
Monks-tr-115	0: 04 :16	29	9.1	69%	67	0: 04 :16	27
	3: 01 :07	27	19.0	30%	1131		
Solar-flare-323	0: 00 :04	51	5.0	90%	18	0: 00 :04	38
	3: 00 :45	43	17.0	61%	977		
BV-OS-376	0: 09 :04	9	3.1	65%	9	0: 09 :04	9
	3: 00 :10	9	6.0	33%	56		
BusVan-445	0: 10 :31	13	3.0	77%	11	0: 10 :31	8
	3: 00 :06	9	5.1	43%	56		
TOTALS		**Gap (same t.)**		**71%**		**1: 44 :36**	
	30: 07 :18	**Gap (end)**		**33%**			

Table 3: Statistical Analysis problems solved to proven optimality by CBC but not by Cplex

Subset 2	Cplex					CBC	
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem. MB**	**Time hh:mm:ss**	**opt.**
CMS 600-0 (4S)	0: 04 :27	592	600	1.35%	18	0: 04 :27	600
	3: 03 :00	594	600	1.01%	770		
CMS 650-0 (4S)	0: 06 :26	638	650	1.88%	20	0: 06 :26	649
	3: 02 :34	646	650	0.62%	480		
CMS 650-1 (4S)	0: 04 :50	647	650	0.46%	7	0: 04 :50	649
	3: 03 :13	648	650	0.31%	904		
CMS 700-1 (4S)	0: 13 :06	686	700	2.04%	58	0: 13 :06	699
	3: 03 :00	691	700	1.30%	1045		
CMS 750-1 (4S)	0: 07 :53	738	750	1.63%	28	0: 07 :53	750
	3: 02 :19	741	750	1.21%	521		
CMS 750-4 (4S)	0: 07 :05	736	750	1.90%	28	0: 07 :05	748
	3: 00 :24	743	750	0.94%	417		
CMS 800-0 (4S)	0: 19 :15	773	800	3.49%	55	0: 19 :15	798
	3: 02 :16	773	800	3.49%	533		
CMS 800-1 (4S)	0: 22 :24	784	800	2.04%	92	0: 22 :24	800
	3: 02 :30	786	800	1.78%	761		
CMS 600-0 (4P)	0: 00 :01	543	600	10.5%	2	0: 00 :04	600
	3: 02 :57	574	600	4.53%	782		
CMS 600-1 (4P)	0: 39 :07	565	600	6.19%	184	0: 39 :07	597
	3: 02 :55	568	600	5.63%	831		
TOTALS		**Gap (same t.)**		**3.6%**		**2: 05 :4.1**	
	33: 25 :10	**Gap (end)**		**2.0%**			

Table 4: Map Labelling problems solved to proven optimality by CBC but not by Cplex

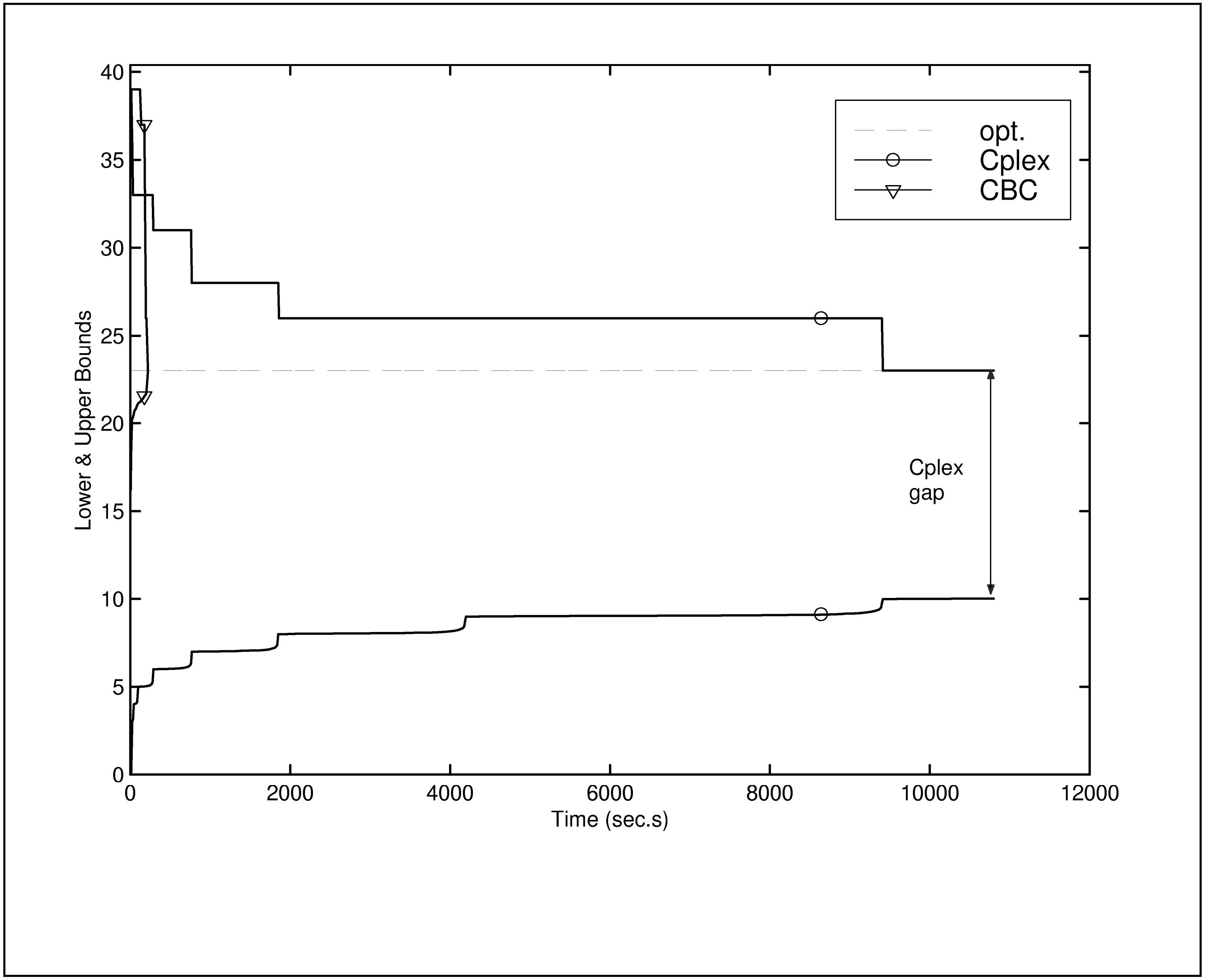

Figure 2: Optimizing the statistical-analysis instance Bridges-132 (minimization problem)

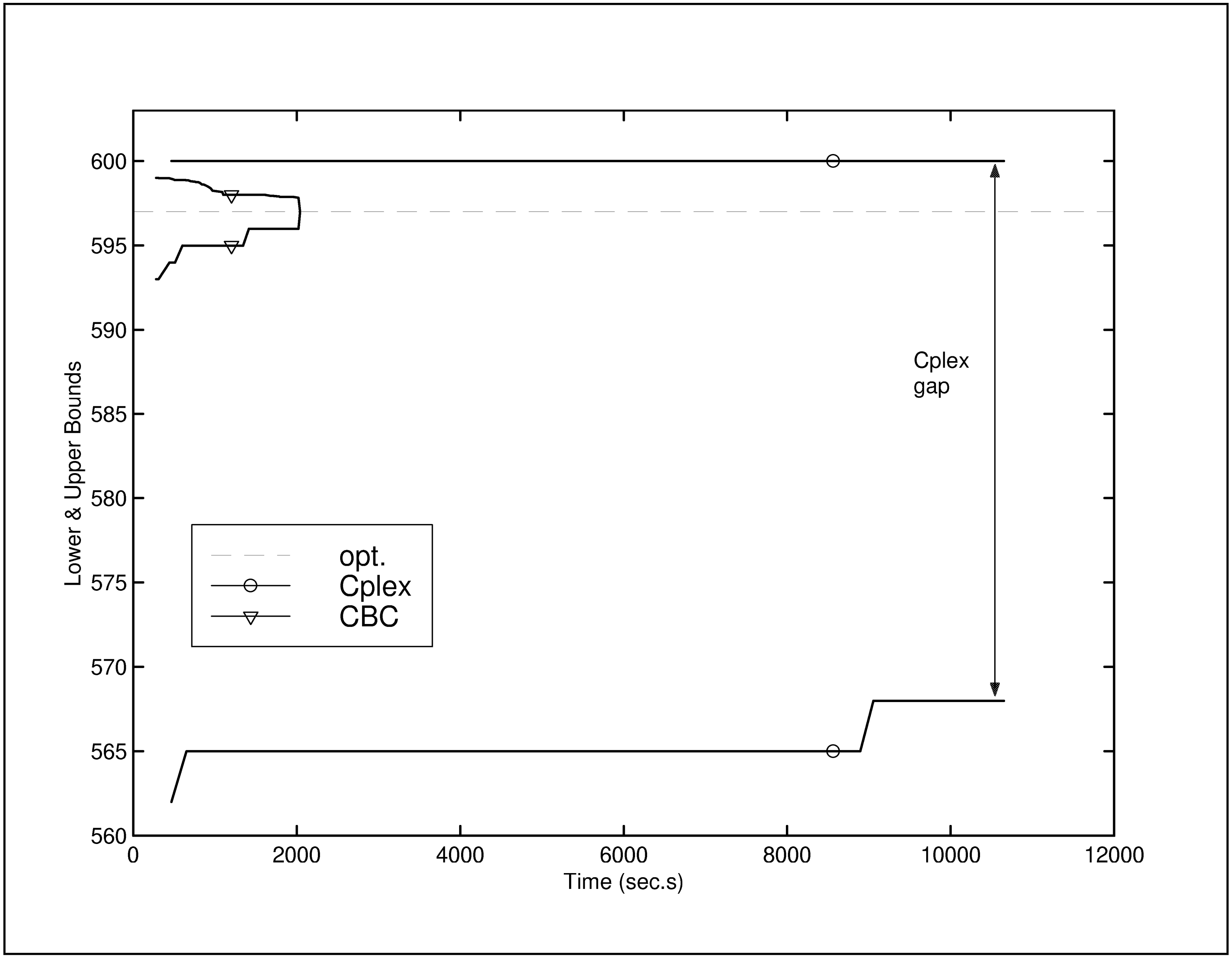

Figure 3: Optimizing the map labelling instance CMS600-1 (4P) (maximization problem)

Subset 3	Cplex				
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem. MB**
Flags-169	3: 00 :19	10	5.0	49.8%	290
Horse-colic-253	3: 00 :15	13	5.0	61.5%	279
Horse-colic-185	3: 00 :16	11	5.0	54.4%	265
Solar-flare-1066	3: 00 :18	273	7.6	97.3%	787
TOTAL	**12: 01 :08**		**Mean Gap**	**65.5%**	—

Subset 3	CBC					
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem. MB**	**$\triangle$Gap %**
Flags-169	3: 02 :46	10	6.50	35.0%	4052	14.8%
Horse-colic-253	3: 02 :59	13	8.91	31.5%	3394	30.0%
Horse-colic-185	3: 01 :25	12	6.33	47.3%	4494	7.1%
Solar-flare-1066	3: 01 :16	284	201.30	29.1%	1423	68.2%
TOTAL	**12: 02 :26**		**Mean Gaps**	**35.7%**	—	**30.0%**

Table 5: Statistical Analysis problems solved to proven optimality by neither codes

Subset 3	Cplex				
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem. MB**
Berlin	3: 06 :43	37	47.8	29.1%	1063
CMS 900-0 (4S)	3: 02 :47	881	900	2.2%	676
CMS 1000-0 (4S)	3: 01 :46	945	1000	5.8%	566
US-Abbrv	3: 01 :18	73	104.8	43.6%	740
CMS 650-0 (4P)	3: 04 :55	611	650	6.4%	764
CMS 650-1 (4P)	3: 02 :51	604	650	7.6%	798
TOTAL	**18: 20 :20**	**Mean Gap**		**15.8%**	—

Subset 3	CBC					
File Name	**Time hh:mm:ss**	**best sol.**	**best bound**	**Gap %**	**mem. MB**	**ΔGap %**
Berlin	3: 03 :43	38	43.0	13.1%	1952	16.0%
CMS 900-0 (4S)	3: 02 :47	897	898.5	0.2%	283	2.0%
CMS 1000-0 (4S)	3: 01 :46	978	998.3	2.1%	509	3.7%
US-Abbrv	3: 01 :18	77	99.7	29.5%	428	14.1%
CMS 650-0 (4P)	3: 05 :17	633	646.9	2.2%	1658	4.2%
CMS 650-1 (4P)	3: 02 :51	638	648.0	1.6%	706	6.0%
TOTAL	**18: 17 :42**	**Mean Gaps**		**8.12%**	—	**7.7%**

Table 6: Map Labelling problems solved to proven optimality by neither codes

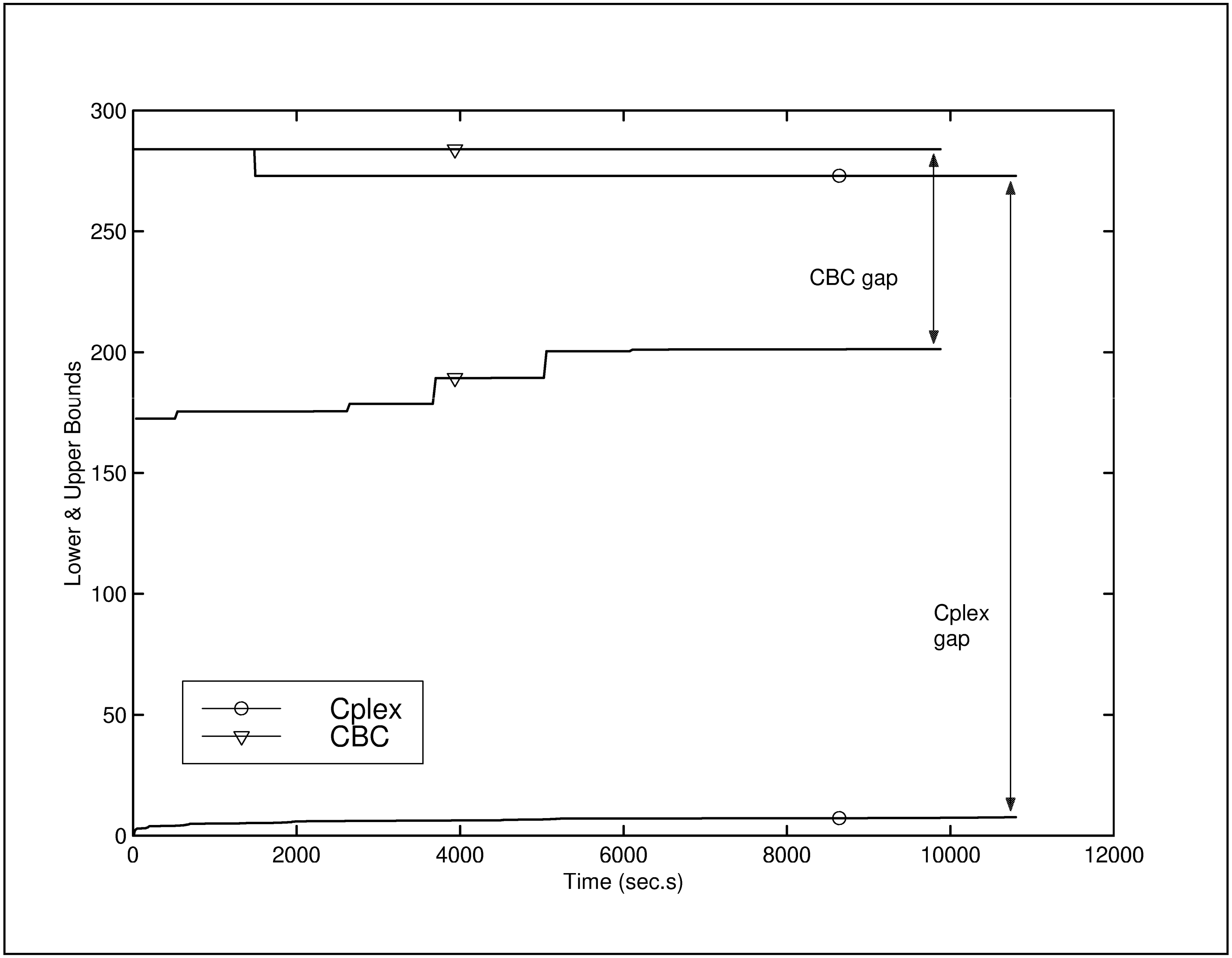

Figure 4: Optimizing the statistical-analysis instance Solar-flare-1066 (minimization problem)

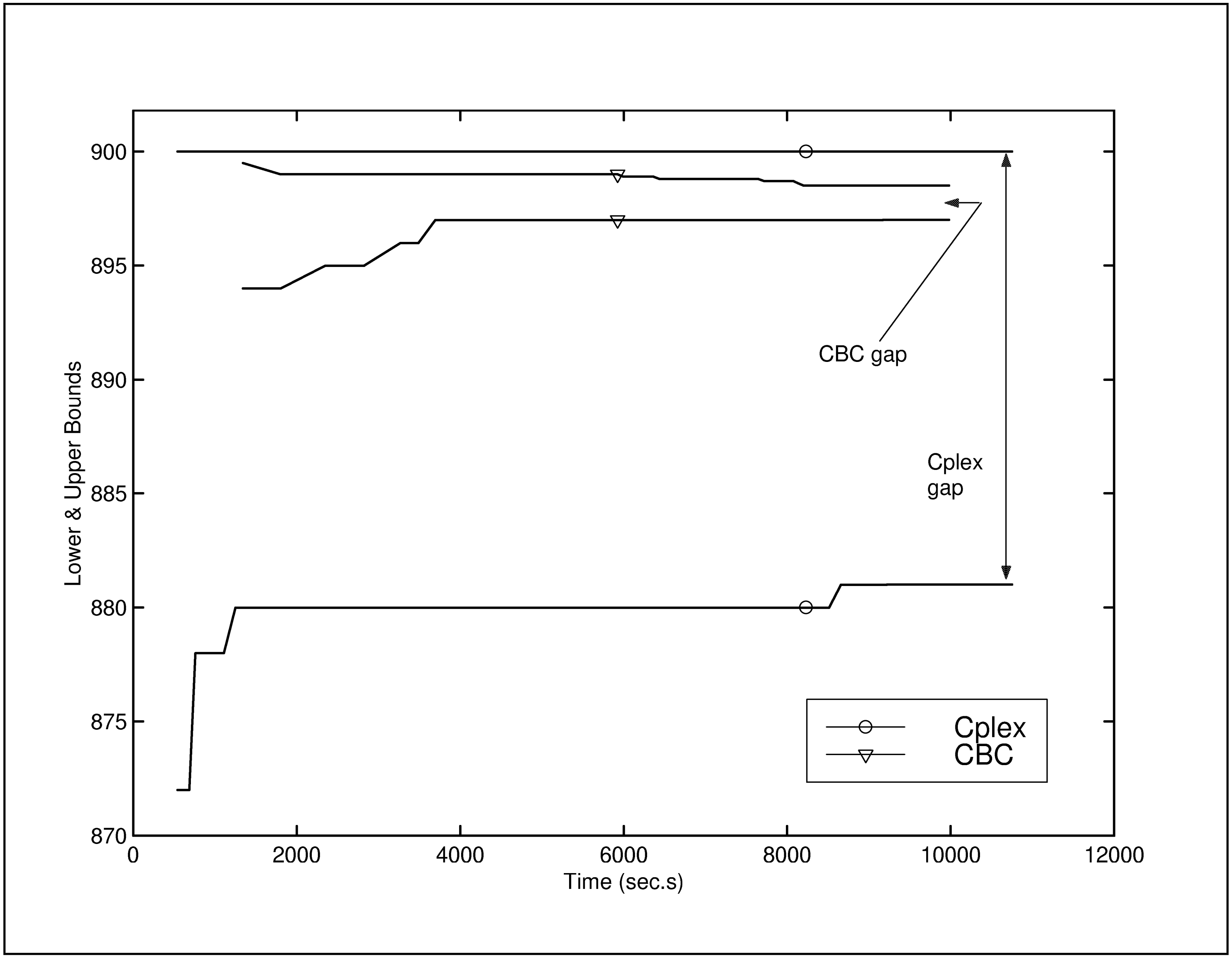

Figure 5: Optimizing the map labelling instance CMS900-0 (4S) (maximization problem)