A Column Generation approach for Pure Parsimony Haplotyping

Dal Sasso Veronica[a] De Giovanni Luigi[a] Labbé Martine[b]

25/06/2015

(a) Dipartimento di Matematica - Università degli Studi di Padova, via Trieste 63, 35121 Padova

(b) GOM, Université Libre de Bruxelles, Bd du Triomphe CP210/01, 1050 Bruxelles, Belgium

Introduction

- Humans are diploid organisms, that is DNA is organized in pairs of chromosomes.

Definition

single nucleotide polymorphism (SNP): site of human genome showing a statistically significant variability within a population.

Example: small portion of a chromosome.

taggtccCtatttCccaggcgcCgtatacttcgacgggTctata
taggtccGtatttAccaggcgcGgtatacttcgacgggTctata

- Almost always, at each SNP site only two nucleotides out of four (A, T, C, G) are observed.
- A SNP can be either homozygous or heterozygous.

Introduction

Definition

Haplotype: it is the set of SNPs on a particular chromosome copy.

Example: haplotypes from the previous chromosome portion: CCCT and GAGT.

Notation: denote with 0 and 1 the two possible nucleotides of every SNP.

Definition

Genotype: it provides information about both the alleles of every SNP, specifying if it is homozygous or heterozygous.

Notation: denote with 0 or 1 homozygous SNPs, with 2 heterozygous sites.

Introduction

Definition

Haplotype: it is the set of SNPs on a particular chromosome copy.

Example: haplotypes from the previous chromosome portion: CCCT and GAGT.

Notation: denote with 0 and 1 the two possible nucleotides of every SNP.

Definition

Genotype: it provides information about both the alleles of every SNP, specifying if it is homozygous or heterozygous.

Notation: denote with 0 or 1 homozygous SNPs, with 2 heterozygous sites.

Introduction

Definition

Compatible haplotype: a haplotype h is compatible with a genotype g if for every site p for which $g_p \neq 2$ we have $g_p = h_p$.

Given two vectors representing two haplotypes h^1 and h^2, we define their sum componentwise as:

$$(h^1 \oplus h^2) = \begin{cases} 0 & \text{if } h^1_p = h^2_p = 0 \\ 1 & \text{if } h^1_p = h^2_p = 1 \\ 2 & \text{if } h^1_p \neq h^2_p \end{cases}$$

Definition

Two haplotypes h^1 and h^2 resolve genotype g if $g = h^1 \oplus h^2$.

Example: $h^1 = 10010$ and $h^2 = 11001$ resolve genotype $g = 12022$.

Introduction

Definition

Compatible haplotype: a haplotype h is compatible with a genotype g if for every site p for which $g_p \neq 2$ we have $g_p = h_p$.

Given two vectors representing two haplotypes h^1 and h^2, we define their sum componentwise as:

$$(h^1 \oplus h^2) = \begin{cases} 0 & \text{if } h^1_p = h^2_p = 0 \\ 1 & \text{if } h^1_p = h^2_p = 1 \\ 2 & \text{if } h^1_p \neq h^2_p \end{cases}$$

Definition

Two haplotypes h^1 and h^2 resolve genotype g if $g = h^1 \oplus h^2$.

Example: $h^1 = 10010$ and $h^2 = 11001$ resolve genotype $g = 12022$.

Introduction

Definition

Compatible haplotype: a haplotype h is compatible with a genotype g if for every site p for which $g_p \neq 2$ we have $g_p = h_p$.

Given two vectors representing two haplotypes h^1 and h^2, we define their sum componentwise as:

$$(h^1 \oplus h^2) = \begin{cases} 0 & \text{if } h^1_p = h^2_p = 0 \\ 1 & \text{if } h^1_p = h^2_p = 1 \\ 2 & \text{if } h^1_p \neq h^2_p \end{cases}$$

Definition

Two haplotypes h^1 and h^2 resolve genotype g if $g = h^1 \oplus h^2$.

Example: $h^1 = 10010$ and $h^2 = 11001$ resolve genotype $g = 12022$.

Introduction

Problem:

- given an individual , obtaining its haplotypes for each chromosome is expensive,
- obtaining its genotypes is cheaper.

But we still need to know the haplotypes: can we deduce them?

- If a genotype has k heterozygous SNPs, there are 2^{k-1} possible pairs of haplotypes that resolve it.

Example: Genotype 12102.

Two pairs: $\{10100, 11101\}, \{11100, 10101\}$

- Given a set of genotypes, there are different sets of haplotypes that resolve it.

We need a criterion to choose the most probable configuration.

Introduction

Problem:

- given an individual , obtaining its haplotypes for each chromosome is expensive,
- obtaining its genotypes is cheaper.

But we still need to know the haplotypes: can we deduce them?

- If a genotype has k heterozygous SNPs, there are 2^{k-1} possible pairs of haplotypes that resolve it.

  Example: Genotype 12102.

  Two pairs: $\{10100, 11101\}, \{11100, 10101\}$

- Given a set of genotypes, there are different sets of haplotypes that resolve it.

We need a criterion to choose the most probable configuration.

Introduction

ASSUMPTION: parsimony principle. Use as few as possible haplotypes to resolve a set of genotypes.

Example: $G = \{20122, 12102, 11122, 02122\}$

$$H' = \{10100, 00111, 11100, 10101, 11101, 11110, 01110, 00101\}$$
$$H'' = \{10100, 00111, 10100, 11101, 11101, 11110, 00111, 01100\}$$

Haplotype Inference by Pure Parsimony problem (HIPP)

Given a set of genotypes G, find a set of haplotypes H such that

- for each genotype $g \in G$, there exists $h^1, h^2 \in H$ such that $g = h^1 \oplus h^2$,
- H has minimum cardinality.

Introduction

ASSUMPTION: parsimony principle. Use as few as possible haplotypes to resolve a set of genotypes.

Example: $G = \{20122, 12102, 11122, 02122\}$

$$H' = \{10100, 00111, 11100, 10101, 11101, 11110, 01110, 00101\}$$
$$H'' = \{10100, 00111, 10100, 11101, 11101, 11110, 00111, 01100\}$$

Haplotype Inference by Pure Parsimony problem (HIPP)

Given a set of genotypes G, find a set of haplotypes H such that

- for each genotype $g \in G$, there exists $h^1, h^2 \in H$ such that $g = h^1 \oplus h^2$,
- H has minimum cardinality.

Different approaches to the solution

- Integer programming formulations of worst-case exponential size, both in the number of variables and constraints (Gusfield (2003), Lancia and Serafini (2008))
  - use variables representing all possible haplotypes;
- integer programming formulations of polynomial size and hybrid formulations (Brown and Harrower (2004, 2005, 2006), Lancia et al. (2004), Bertolazzi et al (2008), Catanzaro et al. (2010))
  - the linear relaxation of these formulations is quite weak
  - addition of valid cuts;
- quadratic, semidefinite programming approaches, of exponential size;
- SAT approaches (Lynce and Marques-Silva(2006), Graça et al. (2011))

References I

[1] De Giovanni L., Labbé M.
Haplotype Inference by Pure Parsimony: a column generation approach.
Personal comunication

[2] Lancia G., Serafini P.
A set-covering approach with column generation for parsimony haplotyping.
INFORMS Journal on Computing, vol. 21 (1), pp. 151-166 (2009)

[3] Lubbecke, M.E.
Column Generation.
Wiley Encyclopedia of Operations Research and Management Science (2010)

[4] Pessoa A., Uchoa E., Poggi de Aragão M. Rodrigues R.
Algorithms over arc-time indexed formulations for single and parallel machine schedule problems.
Report RPEP vol. 8 n. 8, Universitade Federal Fluminense (2008)

First formulation (A) [1]

$$\min \sum_{i=1}^{2m'} x_i \qquad + (m - m') \tag{1}$$

$$s.t. \sum_{i=1}^{m+m'} y_i^k = 2 \quad \forall\, k = 1 \ldots m' \tag{2}$$

$$\sum_{i=1}^{2m'} y_i^k z_{ip} + \sum_{i=2m'+1}^{m+m'} y_i^k g_p^i = 1 \quad \forall\, k = 1 \ldots m',\ p = 1 \ldots n : g_p^k = 2 \tag{3}$$

$$z_{ip} \geq y_i^k \quad \forall\, i = 1 \ldots 2m',\ k = 1 \ldots m',\ p = 1 \ldots n : g_p^k = 1 \tag{4}$$

$$z_{ip} \leq 1 - y_i^k \quad \forall\, i = 1 \ldots 2m',\ k = 1 \ldots m',\ p = 1 \ldots n : g_p^k = 0 \tag{5}$$

$$y_i^k \leq x_i \quad \forall\, i = 1 \ldots 2m',\ k = 1 \ldots m' \tag{6}$$

$$z_{ip} \in \{0, 1\} \quad \forall\, i = 1 \ldots 2m',\ p = 1 \ldots n \tag{7}$$

$$y_i^k \in \{0, 1\} \quad \forall\, i = 1 \ldots m + m' ,\ k = 1 \ldots m' \tag{8}$$

$$x_i \in \{0, 1\} \quad \forall\, i = 1 \ldots 2m' \tag{9}$$

Reformulation using Dantzig-Wolfe decomposition (B)

- Define the set

$$X = \text{conv}\Bigg(\{(z, y, x, w) \in \{0,1\}^{m'(2n+m+m'+2)} |\; w_{ip}^k = y_i^k z_{ip},$$
$$z_{ip} \geq y_i^k \text{ if } g_p^k = 1, z_{ip} \leq 1 - y_i^k \text{ if } g_p^k = 0,$$
$$y_i^k \leq x_i, \sum_{k=1}^{m'} y_i^k \geq x_i \Bigg) \},$$

- X is bounded,
- $\{(z_v, y_v, x_v, w_v) |\; v \in V\}$ is the set of vertices of X,
- if $(z, y, x, w) \in X$ then $(z, y, x, w) = \sum_{v \in V} \theta_v (z_v, y_v, x_v, w_v)$.

Reformulation using Dantzig-Wolfe decomposition (B)

- Define the set

$$X = \text{conv}\Big(\{(z,y,x,w) \in \{0,1\}^{m'(2n+m+m'+2)} \mid w_{ip}^k = y_i^k z_{ip},$$
$$z_{ip} \geq y_i^k \text{ if } g_p^k = 1, z_{ip} \leq 1 - y_i^k \text{ if } g_p^k = 0,$$
$$y_i^k \leq x_i, \sum_{k=1}^{m'} y_i^k \geq x_i \Big)\},$$

- X is bounded,
- $\{(z_v, y_v, x_v, w_v) \mid v \in V\}$ is the set of vertices of X,
- if $(z,y,x,w) \in X$ then $(z,y,x,w) = \sum_{v \in V} \theta_v (z_v, y_v, x_v, w_v)$.

Reformulation using Dantzig-Wolfe decomposition (B)

$$\min \sum_{v \in V} \theta_v \sum_{i=1}^{2m'} (x_v)_i + (m - m') \tag{10}$$

$$s.t. \sum_{v \in V} \theta_v \sum_{i=1}^{m+m'} (y_v)_i^k = 2 \qquad \forall\ k = 1 \ldots m' \tag{11}$$

$$\sum_{v \in V} \theta_v \Big[\sum_{i=1}^{2m'} (w_v)_{ip}^k + \sum_{i=2m'+1}^{m+m'} (y_v)_i^k g_p^i\Big] = 1 \qquad \forall\ k=1\ldots m',\ p=1\ldots n : g_p^k=2 \tag{12}$$

$$\sum_{v \in V} \theta_v = 1 \tag{13}$$

$$\theta_v \in [0,1] \qquad \forall\ v \in V \tag{14}$$

Alternative formulation (C)[1]

What do vertices represent?

- $\{(z_v)_i\}_{i=1,\dots,2m'}$ define $2m'$ haplotypes (not necessarily distincts);
- $\{(y_v)_i\}_{i=1,\dots,m+m'}$ for each i identify the subset of genotypes resolved by the i-th haplotype;
- $\{(x_v)_i\}_{i=1,\dots,2m'}$ counts how many haplotypes are actually used.

Define the pairs $q = (h^q, G^q)$:

- h^q is a haplotype;
- G^q is a subset of genotypes that can be resolved using h^q.

Binary variables λ^q record if the pair q is used ($\lambda^q = 1$) or not ($\lambda^q = 0$) in the solution of our problem.

Alternative formulation (C)[1]

What do vertices represent?

- $\{(z_v)_i\}_{i=1,\ldots,2m'}$ define $2m'$ haplotypes (not necessarily distincts);
- $\{(y_v)_i\}_{i=1,\ldots,m+m'}$ for each i identify the subset of genotypes resolved by the i-th haplotype;
- $\{(x_v)_i\}_{i=1,\ldots,2m'}$ counts how many haplotypes are actually used.

Define the pairs $q = (h^q, G^q)$:

- h^q is a haplotype;
- G^q is a subset of genotypes that can be resolved using h^q.

Binary variables λ^q record if the pair q is used ($\lambda^q = 1$) or not ($\lambda^q = 0$) in the solution of our problem.

Alternative formulation (C)[1]

The formulation obtained:

$$\min \sum_{q \in Q} c^q \lambda^q \qquad + (m - m') \tag{15}$$

$$s.t. \sum_{q: g^k \in G^q} \lambda^q = 2 \quad \forall\, k = 1 \ldots m' \tag{16}$$

$$\sum_{\substack{q: g^k \in G^q \\ h_p^q = 1}} \lambda^q = 1 \quad \forall\, k = 1 \ldots m',\ p = 1 \ldots n : g_p^k = 2 \tag{17}$$

$$\lambda^q \in \{0, 1\} \qquad \forall\, q \in Q \tag{18}$$

Comparison between the formulations

- (A) is non-linear, if we want to solve it using linear programming tecniques we need to linearize it;
- The number of variables increases (linearly) as the number of genotypes or the number of SNPs increase in (A), while in (C) the number of variables increases exponentially;
- The number of constraints increases (also linearly) as the number of genotypes or SNPs increase for (A), (B) and (C)
- (B) and (C) have less constraints than (A)

FOCUS ON

Solving the linear relaxation of formulation (C) with a column generation approach.

Comparison between the formulations

- (A) is non-linear, if we want to solve it using linear programming tecniques we need to linearize it;
- The number of variables increases (linearly) as the number of genotypes or the number of SNPs increase in (A), while in (C) the number of variables increases exponentially;
- The number of constraints increases (also linearly) as the number of genotypes or SNPs increase for (A), (B) and (C)
- (B) and (C) have less constraints than (A)

FOCUS ON

Solving the linear relaxation of formulation (C) with a column generation approach.

Standard column generation [1]

Pricing problem for (C):

PP1) haplotype h is fixed.

$$z(h) = \max \sum_{k=1}^{m'} \left(\bar{\pi}^k + \sum_{\substack{p=1\dots n: g_p^k = 2 \\ h_p = 1}} \bar{\mu}_p^k \right) \chi^k \quad (19)$$

$$s.t.\ h_p \leq 1 - \chi^k \quad \forall\ k = 1\dots m',\ p = 1\dots n : g_p^k = 0 \quad (20)$$

$$h_p \geq \chi^k \quad \forall\ k = 1\dots m',\ p = 1\dots n : g_p^k = 1 \quad (21)$$

$$\chi^k \in \{0,1\} \quad \forall\ k = 1\dots m' \quad (22)$$

- Easily solved by inspection: $\chi^k = 1$ iff g^k is compatible with h and coefficient in brackets is ≥ 0.
- Pair $q^\star = (h, G^{q^\star})$ to be added if $z(h) > 0$

Standard column generation [1]

Pricing problem for (C):

PP1) haplotype h is fixed.

$$z(h) = \max \sum_{k=1}^{m'} \left(\bar{\pi}^k + \sum_{\substack{p=1\dots n: g_p^k=2 \\ h_p=1}} \bar{\mu}_p^k \right) \chi^k \tag{19}$$

$$s.t.\ h_p \leq 1 - \chi^k \qquad \forall\ k = 1 \dots m',\ p = 1 \dots n : g_p^k = 0 \tag{20}$$

$$h_p \geq \chi^k \qquad \forall\ k = 1 \dots m',\ p = 1 \dots n : g_p^k = 1 \tag{21}$$

$$\chi^k \in \{0, 1\} \qquad \forall\ k = 1 \dots m' \tag{22}$$

- Easily solved by inspection: $\chi^k = 1$ iff g^k is compatible with h and coefficient in brackets is ≥ 0.
- Pair $q^\star = (h, G^{q^\star})$ to be added if $z(h) > 0$

Standard column generation [1]

PP2) haplotype h is not fixed.

$$z = \max \sum_{k=1}^{m'} \left(\bar{\pi}^k + \sum_{p=1\dots n : g_p^k = 2} \bar{\mu}_p^k \, \zeta_p \right) \chi^k \tag{23}$$

$$s.t.\ \zeta_p \leq 1 - \chi^k \qquad \forall\ k = 1\dots m',\ p = 1\dots n : g_p^k = 0 \tag{24}$$

$$\zeta_p \geq \chi^k \qquad \forall\ k = 1\dots m',\ p = 1\dots n : g_p^k = 1 \tag{25}$$

$$\chi^k, \zeta_p \in \{0, 1\} \qquad \forall\ k = 1\dots m',\ p = 1\dots n \tag{26}$$

- It's a quadratic pricing problem.
- Pair $q^\star$ to be added is found if $z > 1$.

Standard column generation [1]

PP2) haplotype h is not fixed.

$$z = \max \sum_{k=1}^{m'} \left(\bar{\pi}^k + \sum_{p=1\dots n : g_p^k = 2} \bar{\mu}_p^k \, \zeta_p \right) \chi^k \tag{23}$$

$$s.t.\ \zeta_p \leq 1 - \chi^k \qquad \forall\, k = 1 \dots m',\ p = 1 \dots n : g_p^k = 0 \tag{24}$$

$$\zeta_p \geq \chi^k \qquad \forall\, k = 1 \dots m',\ p = 1 \dots n : g_p^k = 1 \tag{25}$$

$$\chi^k, \zeta_p \in \{0, 1\} \qquad \forall\, k = 1 \dots m',\ p = 1 \dots n \tag{26}$$

- It's a quadratic pricing problem.
- Pair $q^\star$ to be added is found if $z > 1$.

Standard column generation [1]

Outline of the algorithm [CG]:

1) choose an initial feasible solution (starting set of variables)
2) solve the Restricted Master Problem (RMP) and get the current value $\tilde{v}$ and solution $\tilde{\lambda}$;
3) get the associated dual variables $\bar{\pi}$, $\bar{\mu}$;
4) solve the Pricing Problem:
   - solve PP1 for every fixed haplotype h. If a suitable $q^\star$ is found, then add it to RMP. Go back to 2). If not:
   - use a local search. If a suitable $q^\star$ is found, add it to RMP. Go back to 2). Otherwise:
   - solve PP2.
5) if PP2 does not find a suitable $q^\star$, **STOP**. Otherwise, add the new variable to RMP and go back to 2)

Computational challenges

- Tailing-off effect: only little progress is made near the optimal solution
- Highly degenerate problems: difficulty in recognising an optimal solution

$\Longrightarrow$ Find a lower bound on the optimal solution as an early termination condition.

- Instability: the dual variables do not smoothly converge to the optimal solution

$\Longrightarrow$ Use a stabilization technique: convex combination of dual variables with previous values.

Computational challenges

- Tailing-off effect: only little progress is made near the optimal solution
- Highly degenerate problems: difficulty in recognising an optimal solution

$\Longrightarrow$ Find a lower bound on the optimal solution as an early termination condition.

- Instability: the dual variables do not smoothly converge to the optimal solution

$\Longrightarrow$ Use a stabilization technique: convex combination of dual variables with previous values.

Lagrangian lower bound for (B)

- Define $\Theta = \{\theta \in [0,1]^{|V|} \mid \theta_v \geq 0, \sum_{v \in V} \theta_v = 1\}$
- Define the Lagrangian function

$$L(\pi,\mu) = \min_{\theta \in \Theta} \Big\{ \sum_{v \in V} \theta_v \sum_{i=1}^{2m'} (x_v)_i - \sum_{k=1}^{m'} \pi^k \Big(\sum_{v \in V} \theta_v \sum_{i=1}^{m+m'} (y_v)_i^k - 2 \Big) -$$

$$- \sum_{k,p:g_p^k=2} \mu_p^k \Big(\sum_{v \in V} \theta_v \Big[\sum_{i=1}^{2m'} (w_v)_{ip}^k + \sum_{i=2m'+1}^{m+m'} (y_v)_i^k g_p^i \Big] - 1 \Big) \Big\} =$$

$$= v_D(\pi,\mu) + \min_{v \in V} \Big\{ \sum_{i=1}^{2m'} (x_v)_i - \sum_{i=1}^{m+m'} \sum_{k=1}^{m'} (y_v)_i^k - \sum_{i=1}^{2m'} \sum_{k,p:g_p^k=2} \mu_p^k (w_v)_{ip}^k -$$

$$- \sum_{i=2m'+1}^{m+m'} \mu_p^k (y_v)_i^k g_p^i \Big\} =$$

$$= v_D(\pi,\mu) + (m+m')(c - v_{PP}(\pi,\mu))$$

Lagrangian lower bound for (C)

- Add a redundant constraint to formulation (C) acting as an upper bound on the optimal solution.
- M is an appropriate value: equal to the current objective value of [CG]).

Formulation (C):

$$\min \sum_{q \in Q} c^q \lambda^q \qquad + (m - m') \tag{27}$$

$$s.t. \sum_{q: g^k \in G^q} \lambda^q = 2 \qquad \forall\, k = 1 \dots m' \tag{28}$$

$$\sum_{\substack{q: g^k \in G^q \\ h_p^q = 1}} \lambda^q = 1 \qquad \forall\, k = 1 \dots m',\; p = 1 \dots n : g_p^k = 2 \tag{29}$$

$$\sum_{q \in Q} \lambda^q \leq M \tag{30}$$

$$\lambda^q \in [0, 1] \qquad \forall\, q \in Q \tag{31}$$

Lagrangian lower bound for (C)

- Define $\Lambda = \{\lambda \in [0,1]^{|Q|} : \lambda^q \geq 0, \sum_{q \in Q} \lambda^q \leq M\}$
- Define the Lagrangian function

$$L(\pi,\mu) = \min_{\lambda \in \Lambda} \Big\{ \sum_{q \in Q} c^q \lambda^q - \sum_{k} \pi^k \Big(\sum_{q : g^k \in G^q} \lambda^q - 2 \Big) - \sum_{k,p : g^k_p = 2} \mu^k_p \Big(\sum_{\substack{q : g^k \in G^q, \\ h^q_p = 1}} \lambda^q - 1 \Big) \Big\} =$$

$$= v_D(\pi,\mu) + \min_{\lambda \in \Lambda} \Big\{ \sum_{q \in Q} \Big[c^q - \sum_{k : g^k \in G^q} \pi^k - \sum_{k : g^k \in G^q} \sum_{p : g^k_p = 2,\ h^q_p = 1} \mu^k_p \Big] \lambda^q \Big\} =$$

$$= v_D(\pi,\mu) + M \min_{q \in Q} \Big\{ c_q - \sum_{k : g^k \in G^q} \pi^k - \sum_{k : g^k \in G^q} \sum_{p : g^k_p = 2,\ h^q_p = 1} \mu^k_p \Big\} =$$

$$= v_D(\pi,\mu) + M(c - v_{PP}(\pi,\mu))$$

- Better lower bound: $M \leq m + m'$

Lagrangian lower bound for (C)

- Define $\Lambda = \{\lambda \in [0,1]^{|Q|} : \ \lambda^q \geq 0, \sum_{q \in Q} \lambda^q \leq M\}$
- Define the Lagrangian function

$$L(\pi,\mu) = \min_{\lambda \in \Lambda} \Big\{ \sum_{q \in Q} c^q \lambda^q - \sum_k \pi^k \big(\sum_{q : g^k \in G^q} \lambda^q - 2 \big) - \sum_{k,p : g^k_p = 2} \mu^k_p \big(\sum_{\substack{q : g^k \in G^q, \\ h^q_p = 1}} \lambda^q - 1 \big) \Big\} =$$

$$= v_D(\pi,\mu) + \min_{\lambda \in \Lambda} \Big\{ \sum_{q \in Q} [c^q - \sum_{k : g^k \in G^q} \pi^k - \sum_{k : g^k \in G^q} \sum_{p : g^k_p = 2, \ h^q_p = 1} \mu^k_p] \lambda^q \Big\} =$$

$$= v_D(\pi,\mu) + M \min_{q \in Q} \Big\{ c_q - \sum_{k : g^k \in G^q} \pi^k - \sum_{k : g^k \in G^q} \sum_{p : g^k_p = 2, \ h^q_p = 1} \mu^k_p \Big\} =$$

$$= v_D(\pi,\mu) + M(c - v_{PP}(\pi,\mu))$$

- Better lower bound: $M \leq m + m'$

Improved algorithm

- $L(\pi, \mu) \leq z^{OPT}$ for all (π, μ) feasible
- Use a lower bound as an early termination condition
- Compute lower bound when solving exact PP
- Consider the dual solution of RMP: lower bound provided without effort
- Algorithm [CG] ends if
  - no suitable variable is found to be added to RMP,
  - the gap between the primal objective value and the lower bound is less than a value ϵ.

Convex combination with previous dual solutions

Basic idea

A stabilization method is used to bound the dual variables values.

Examples of stabilization methods:

- Interior point stabilization,
- Box-step method,
- Bundle methods,
- Convex combination with previous dual solutions

Merits of this procedure:

- helps avoiding too large steps in the dual space
- easy to implement: do not need to change the RMP

The stabilized pricing algorithm

1) set $0 < \alpha < 1$, initialize $(\bar{\pi}, \bar{\mu}, \bar{\nu}) = 0$,
2) solve the RMP and get the objective value z_{RM} and the dual variables associated $(\pi_{RM}, \mu_{RM}, \nu_{RM})$,
3) compute $(\pi_{ST}, \mu_{ST}, \nu_{ST}) = \alpha(\pi_{RM}, \mu_{RM}, \nu_{RM}) + (1 - \alpha)(\bar{\pi}, \bar{\mu}, \bar{\nu})$ to be used in the pricing problem,
4) if $q^\star$ violates a dual constraint w.r.t $(\pi_{RM}, \mu_{RM}, \nu_{RM})$, then add it to the RMP,
5) if the $q^\star$ found is the optimal solution of PP2 and $LB(\pi_{ST}, \mu_{ST}, \nu_{ST}) > LB(\bar{\pi}, \bar{\mu}, \bar{\nu})$, then update $(\bar{\pi}, \bar{\mu}, \bar{\nu}) = (\pi_{ST}, \mu_{ST}, \nu_{ST})$,
6) iterate until $z_{RM} - LB(\bar{\pi}, \bar{\mu}, \bar{\nu}) < \epsilon$.

Convergence of the procedure

Lemma

If the solution of the pricing problem with stabilized coefficients does not give a variable that violates a dual constraint w.r.t. $(\pi_{RM}, \mu_{RM}, \nu_{RM})$, *then*

$$LB(\pi_{ST}, \mu_{ST}, \nu_{ST}) > LB(\bar{\pi}, \bar{\mu}, \bar{\nu}) + \alpha(z_{RM} - LB(\bar{\pi}, \bar{\mu}, \bar{\nu}))$$

A misprice then is not a loss of time:

- it guarantees an improvement on the lower bound,
- the gap $z_{RM} - LB(\bar{\pi}, \bar{\mu}, \bar{\nu})$ is reduced of at least a factor $1/(1-\alpha)$,
- the stability center changes, so that we do not get stuck in a non-optimal solution.

Instances

- Brown and Harrower instances
- Real data and random instances

Instance	# SNPs	# genotypes	#fixed	% av. het. SNPs
1	10	50	11	39.80
2	30	36	4	25.10
3	30	20	4	39.67
4	30	12	3	33.06
5	30	7	1	55.71
6	50	10	2	52.80
7	50	5	2	37.60

- implementation: C++ with SCIP 3.1 and Cplex 12.4 on an Intel Core i7 2GHz
- Set parameter for stabilization: $\alpha = 0.2$
- Set tolerance for Lagrangian bound: $\epsilon = 0.1$

Standard and Stabilized Column Generation

Column Generation

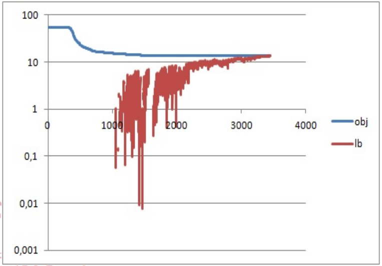
Instance	time (s)	# PP2	$z_{LP} - LB$	%Opt-PP2
1	297.11	367	0.02	35.05
2	37247,87	3447	0.04	29.23
3	21566,72	5744	0.00	19.16
4	1740.95	1905	0.09	34.10
5	6316.01	2801	0.02	49.82
6	36457.49	22105	0.17	0.01
7	1041.92	601	0.41	0.33

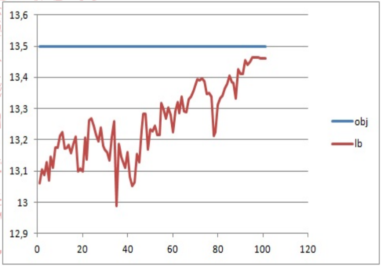
Stabilized Column Generation

Instance	time (s)	#PP2	$z_{LP} - LB$	%Opt-PP2
1	452.94	268	0.09	48.5
2	18226.36	1562	0.09	31.82
3	6825.06	1244	0.09	6.91
4	1109.67	596	0.10	22.65
5	753.55	462	0.10	15.58
6	7197.36	2149	0.08	1.58
7	147.09	268	0.09	13.06

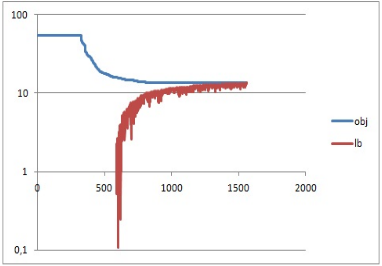
Standard and Stabilized Column Generation

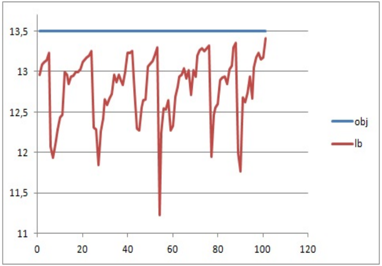
Column Generation





Stabilized Column Generation



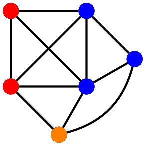


Conclusions and further work

- Column generation was necessary to handle the great number of variables. Anyway, there are issues to be overcome
- One optimal solution is still found quite early if compared with satisfying a termination condition
  $\Longrightarrow$ look for a different lower bound that dominate the current one
- Provide a better solution to start the column generation procedure

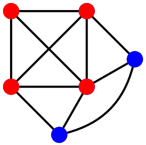
Conclusions and further work

- Column generation was necessary to handle the great number of variables. Anyway, there are issues to be overcome
- One optimal solution is still found quite early if compared with satisfying a termination condition
  $\Longrightarrow$ look for a different lower bound that dominate the current one
- Provide a better solution to start the column generation procedure



Conclusions and further work

- Column generation was necessary to handle the great number of variables. Anyway, there are issues to be overcome
- One optimal solution is still found quite early if compared with satisfying a termination condition
  $\Longrightarrow$ look for a different lower bound that dominate the current one
- Provide a better solution to start the column generation procedure



Thanks for the attention