"Non-commutative discrete optimization"

Alexei Miasnikov
(Stevens Institute)

Symbolic Computations and Post-Quantum Cryptography
Web Seminar
March 21st, 2013,

(based on joint work with A.Nikolaev and A.Ushakov)

Outline

- What is non-commutative discrete optimization?
- Knapsack problems in groups.
- More open problems.

Non-commutative discrete optimization

Non-commutative discrete (combinatorial) optimization concerns with complexity of the classical discrete optimization (DO) problems stated in a very general form - for non-commutative groups.

Non-commutative discrete optimization

DO problems concerning integers (subset sum, knapsack problem, etc.) make perfect sense when the group of additive integers is replaced by an arbitrary (non-commutative) group G.

The classical subset sum problem (**SSP**): Given $a_1, \ldots, a_k, a \in \mathbb{Z}$ decide if $\varepsilon_1 a_1 + \ldots + \varepsilon_k a_k = a$ for some $\varepsilon_1, \ldots, \varepsilon_k \in \{0, 1\}$.

SSP for a group G:

Given $g_1, \ldots, g_k, g \in G$ decide if $g_1^{\varepsilon_1} \ldots g_k^{\varepsilon_k} = g$ for some $\varepsilon_1, \ldots, \varepsilon_k \in \{0, 1\}$.

Elements in G are given as words in a fixed set of generators of G.

Non-commutative discrete optimization

DO problems concerning integers (subset sum, knapsack problem, etc.) make perfect sense when the group of additive integers is replaced by an arbitrary (non-commutative) group G.

The classical subset sum problem (**SSP**): Given $a_1, \ldots, a_k, a \in \mathbb{Z}$ decide if $\varepsilon_1 a_1 + \ldots + \varepsilon_k a_k = a$ for some $\varepsilon_1, \ldots, \varepsilon_k \in \{0, 1\}$.

SSP for a group G:

Given $g_1, \ldots, g_k, g \in G$ decide if $g_1^{\varepsilon_1} \ldots g_k^{\varepsilon_k} = g$ for some $\varepsilon_1, \ldots, \varepsilon_k \in \{0, 1\}$.

Elements in G are given as words in a fixed set of generators of G.

The classical lattice problems are about subgroups (integer lattices) of the additive groups $\mathbb{Z}^n$ or $\mathbb{Q}^n$, their non-commutative versions deal with arbitrary finitely generated subgroups of a group G.

The shortest vector problem (**SVP**): Find a shortest vector in a given lattice L of $\mathbb{Z}^n$ (or $\mathbb{Q}^n$).

SVP for a group G:

Find a shortest element (in the word metric) in a subgroup of G generated by elements $g_1, \ldots, g_k \in G$.

The classical lattice problems are about subgroups (integer lattices) of the additive groups $\mathbb{Z}^n$ or $\mathbb{Q}^n$, their non-commutative versions deal with arbitrary finitely generated subgroups of a group G.

The shortest vector problem (**SVP**): Find a shortest vector in a given lattice L of $\mathbb{Z}^n$ (or $\mathbb{Q}^n$).

SVP for a group G:

Find a shortest element (in the word metric) in a subgroup of G generated by elements $g_1, \ldots, g_k \in G$.

The classical lattice problems are about subgroups (integer lattices) of the additive groups $\mathbb{Z}^n$ or $\mathbb{Q}^n$, their non-commutative versions deal with arbitrary finitely generated subgroups of a group G.

The shortest vector problem (**SVP**): Find a shortest vector in a given lattice L of $\mathbb{Z}^n$ (or $\mathbb{Q}^n$).

SVP for a group G:

Find a shortest element (in the word metric) in a subgroup of G generated by elements $g_1, \ldots, g_k \in G$.

Non-commutative discrete optimization

The travelling salesman problem, the Steiner tree problem, the Hamiltonian circuit problem, - all make sense for arbitrary finite subsets of vertices in a given Cayley graph of a non-commutative infinite group (with the word metric).

Let G be a group generated by a finite set X and $Cay(G, X)$ the Cayley graph of G.

Traveling Salesman Problem in G:

Given a finite set of vertices $v_1, \ldots, v_n \in Cay(G, X)$ find a closed tour of minimal total length (in the word metric) that visits all the vertices once.

Non-commutative discrete optimization

The travelling salesman problem, the Steiner tree problem, the Hamiltonian circuit problem, - all make sense for arbitrary finite subsets of vertices in a given Cayley graph of a non-commutative infinite group (with the word metric).

Let G be a group generated by a finite set X and $Cay(G, X)$ the Cayley graph of G.

Traveling Salesman Problem in G:

Given a finite set of vertices $v_1, \ldots, v_n \in Cay(G, X)$ find a closed tour of minimal total length (in the word metric) that visits all the vertices once.

This list of examples can be easily extended, but the point here is that many classical DO problems have natural and interesting non-commutative versions.

All these classical problems are $\mathbb{NP}$-complete.

Complexity of their non-commutative analogs depends on the group.

This list of examples can be easily extended, but the point here is that many classical DO problems have natural and interesting non-commutative versions.

All these classical problems are $\mathbb{NP}$-complete.

Complexity of their non-commutative analogs depends on the group.

There are three principle Knapsack type problems in groups: subset sum, knapsack, and submonoid membership.

We have mentioned already the subset sum problem **SSP** in groups. The classical **SSP** is the most basic **NP**-complete problem, it became famous after Merkle-Hellman's cryptosystem.

There are three principle Knapsack type problems in groups: subset sum, knapsack, and submonoid membership.

We have mentioned already the subset sum problem **SSP** in groups. The classical **SSP** is the most basic **NP**-complete problem, it became famous after Merkle-Hellman's cryptosystem.

The knapsack problem (KP) for G:

Given $g_1, \ldots, g_k, g \in G$ decide if $g =_G g_1^{\varepsilon_1} \ldots g_k^{\varepsilon_k}$ for some non-negative integers $\varepsilon_1, \ldots, \varepsilon_k$.

There are minor variations of this problem, for instance, integer **KP**, when ε_i are arbitrary integers. They are all similar, we omit them here.

The subset sum problem sometimes is called $0-1$ knapsack.

The knapsack problems in groups is closely related to the big powers method, which appeared long before any complexity considerations (Baumslag, 1962).

The method shaped up as a basic tool in the study of

- equations in free or hyperbolic groups,
- in algebraic geometry over groups groups,
- completions and group actions,
- became a routine in the theory of hyperbolic groups (in the form of properties of quasideodesics).

Submonoid membership problem in groups

The third problem is equivalent to **KP** in the classical (abelian) case, but not in general, it is of prime interest in algebra:

Submonoid membership problem (SMP):

Given a finite set $A = \{g_1, \ldots, g_k, g\}$ of elements of G decide if g belongs to the submonoid generated by A, i.e., if $g = g_{i_1}, \ldots, g_{i_s}$ for some $g_{i_j} \in A$.

If the set A is closed under inversion then we have the subgroup membership problem in G.

Algorithmic set-up

G is a group generated by a set $X \subseteq G$.

Elements in G are given as group words over X.
If X is finite then the size of a word g in $X^{\pm}$ is its length $|g|$.

The size of a tuple of words $g_1, \ldots, g_k$ is the total sum of the lengths $|g_1| + \ldots + |g_k|$.

Algorithmic set-up

If the generating set X is infinite, then the size of a letter $x \in X$ is not necessarily equal to 1, it depends on how we represent elements of X.

We always assume that there is an efficient injective function $\nu : X \to \{0,1\}^*$ which encodes elements in X by binary strings.

In this case for $x \in X$ we define:

$$\mathrm{size}(x) = |\nu(x)|,$$

for a word $g = x_1 \ldots x_n$ with $x_i \in X$

$$\mathrm{size}(g) = \mathrm{size}(x_1) + \ldots + \mathrm{size}(x_n),$$

for a tuple of words $(g_1, \ldots, g_k)$

$$\mathrm{size}(g_1, \ldots, g_k) = \mathrm{size}(g_1) + \ldots + \mathrm{size}(g_k).$$

Algorithmic set-up

If the generating set X is infinite, then the size of a letter $x \in X$ is not necessarily equal to 1, it depends on how we represent elements of X.

We always assume that there is an efficient injective function $\nu : X \to \{0,1\}^*$ which encodes elements in X by binary strings.

In this case for $x \in X$ we define:

$$\mathrm{size}(x) = |\nu(x)|,$$

for a word $g = x_1 \ldots x_n$ with $x_i \in X$

$$\mathrm{size}(g) = \mathrm{size}(x_1) + \ldots + \mathrm{size}(x_n),$$

for a tuple of words $(g_1, \ldots, g_k)$

$$\mathrm{size}(g_1, \ldots, g_k) = \mathrm{size}(g_1) + \ldots + \mathrm{size}(g_k).$$

It makes sense to consider the bounded versions of **KP** and **SMP**, they are always decidable in groups with decidable word problem.

The bounded knapsack problem (BKP) for G:

decide, when given $g_1, \ldots, g_k, g \in G$ and $1^m \in \mathbb{N}$, if $g =_G g_1^{\varepsilon_1} \ldots g_k^{\varepsilon_k}$ for some $\varepsilon_i \in \{0, 1, \ldots, m\}$.

This problem is **P**-time equivalent to **SSP** in G.

It makes sense to consider the bounded versions of **KP** and **SMP**, they are always decidable in groups with decidable word problem.

The bounded knapsack problem (BKP) for G:

decide, when given $g_1, \ldots, g_k, g \in G$ and $1^m \in \mathbb{N}$, if $g =_G g_1^{\varepsilon_1} \ldots g_k^{\varepsilon_k}$ for some $\varepsilon_i \in \{0, 1, \ldots, m\}$.

This problem is **P**-time equivalent to **SSP** in G.

The bounded **SMP** in G is very interesting in its own right.

Bounded submonoid membership problem (BSMP) for G:

Given $g_1, \ldots g_k, g \in G$ and $1^m \in \mathbb{N}$ (in unary) decide if g is equal in G to a product of the form $g = g_{i_1} \cdots g_{i_s}$, where $g_{i_1}, \ldots, g_{i_s} \in \{g_1, \ldots, g_k\}$ and $s \leq m$.

In search variations we are asked to find a particular solution.

We will discuss later the optimization version of search problems, when one has to find a solution under some optimal restrictions.

Search variations

In search variations we are asked to find a particular solution.

We will discuss later the optimization version of search problems, when one has to find a solution under some optimal restrictions.

Classical **SSP**

As we mentioned the classical **SSP** is **NP**-complete when the numbers are given in binary.

But if the numbers in **SSP** are given in unary, then the problem is in **P** (the problem is pseudo-polynomial).

How one explain this from the group-theoretic view-point?

Classical **SSP**

As we mentioned the classical **SSP** is **NP**-complete when the numbers are given in binary.

But if the numbers in **SSP** are given in unary, then the problem is in **P** (the problem is pseudo-polynomial).

How one explain this from the group-theoretic view-point?

- $\mathbb{Z}$ is generated by $\{1\}$. Then **SSP**$(\mathbb{Z}, \{1\})$ is linear-time equivalent to the classical **SSP** in which numbers are given in unary. In particular, **SSP**$(\mathbb{Z}, \{1\})$ is in **P**.
- $\mathbb{Z}$ is generated by $X = \{x_n = 2^n \mid n \in \mathbb{N}\}$. Fix an encoding $\nu : X^{\pm 1} \to \{0,1\}^*$ such that $size(x_n)$ is about n. Then **SSP**$(\mathbb{Z}, X)$ is **P**-time equivalent to its classical version where the numbers are given in the binary form. In particular, **SSP**$(\mathbb{Z}, X)$ is **NP**-complete.

- $\mathbb{Z}$ is generated by $\{1\}$. Then **SSP**$(\mathbb{Z}, \{1\})$ is linear-time equivalent to the classical **SSP** in which numbers are given in unary. In particular, **SSP**$(\mathbb{Z}, \{1\})$ is in **P**.
- $\mathbb{Z}$ is generated by $X = \{x_n = 2^n \mid n \in \mathbb{N}\}$. Fix an encoding $\nu : X^{\pm 1} \to \{0, 1\}^*$ such that $size(x_n)$ is about n. Then **SSP**$(\mathbb{Z}, X)$ is **P**-time equivalent to its classical version where the numbers are given in the binary form. In particular, **SSP**$(\mathbb{Z}, X)$ is **NP**-complete.

Let $G = \mathbb{Z}^\omega$, $E = \{\mathbf{e}_i\}_{i \in \mathbb{N}}$ is the standard basis for $\mathbb{Z}^\omega$.

We fix an encoding $\nu : E^{\pm 1} \to \{0,1\}^*$ for the generating set E defined by:

$$\begin{cases} \mathbf{e}_i & \stackrel{\nu}{\mapsto} & 0101(00)^i 11, \\ -\mathbf{e}_i & \stackrel{\nu}{\mapsto} & 0100(00)^i 11. \end{cases}$$

Theorem

SSP$(\mathbb{Z}^\omega, E)$ is **NP**-complete.

Proof. The following $\mathbb{NP}$-complete problem is **P**time reducible to **SSP**$(\mathbb{Z}^\omega, E)$.

Zero-one equation problem: Given a zero-one matrix $A \in Mat(n, \mathbb{Z})$ decide if there exists a zero-one vector $x \in \mathbb{Z}^n$ satisfying $A \cdot x = 1_n$, or not.

Crucial lemma

To formulate the following results put

$$\mathcal{P} = \{\mathbf{SSP}, \mathbf{KP}, \mathbf{SMP}, \mathbf{BKP}, \mathbf{BSMP}\}.$$

Ptime embeddings

Let G_i be a group generated by a set X_i with an encoding ν_i, $i = 1, 2$. If

$$\phi : G_1 \to G_2$$

is a **P**-time computable embedding relative to $(X_1, \nu_1), (X_2, \nu_2)$ then $\mathbf{\Pi}(G_1, X_1)$ is **P**-time reducible to $\mathbf{\Pi}(G_2, X_2)$ for any problem $\mathbf{\Pi} \in \mathcal{P}$.

If X_1, X_2 are finite then any embedding $\phi : G_1 \to G_2$ is a **P**-time computable.

In particular, any problem from $\mathcal{P}$ is **P**time equivalent upon changing finite generating sets.

Crucial lemma

To formulate the following results put

$$\mathcal{P} = \{\mathbf{SSP}, \mathbf{KP}, \mathbf{SMP}, \mathbf{BKP}, \mathbf{BSMP}\}.$$

Ptime embeddings

Let G_i be a group generated by a set X_i with an encoding ν_i, $i = 1, 2$. If

$$\phi : G_1 \to G_2$$

is a **P**-time computable embedding relative to $(X_1, \nu_1), (X_2, \nu_2)$ then $\mathbf{\Pi}(G_1, X_1)$ is **P**-time reducible to $\mathbf{\Pi}(G_2, X_2)$ for any problem $\mathbf{\Pi} \in \mathcal{P}$.

If X_1, X_2 are finite then any embedding $\phi : G_1 \to G_2$ is a **P**-time computable.

In particular, any problem from $\mathcal{P}$ is **P**time equivalent upon changing finite generating sets.

Examples

The following groups have **NP**-complete **SSP**:

(a) Free metabelian non-abelian groups of finite rank.

(b) Wreath product $\mathbb{Z} \wr \mathbb{Z}$.

Let M_n be a free metabelian group with basis $X = \{x_1, \ldots, x_n\}$, where $n \geq 2$. A map

$$e_i \to x_1^{-i}[x_2, x_1]x_1^i \quad (\text{for } i \in \mathbb{N})$$

gives a **P**-time embedding of $\mathbb{Z}^\omega$ into M_n.

Let $G = \langle a \rangle \; wr \; \langle t \rangle$. A map $e_i \to t^{-i}at^i, \quad i \in \mathbb{N}$ gives a **P**-time embedding of $\mathbb{Z}^\omega$ into G.

Examples

The following groups have **NP**-complete **SSP**:

(a) Free metabelian non-abelian groups of finite rank.

(b) Wreath product $\mathbb{Z} \wr \mathbb{Z}$.

Let M_n be a free metabelian group with basis $X = \{x_1, \ldots, x_n\}$, where $n \geq 2$. A map

$$e_i \to x_1^{-i}[x_2, x_1]x_1^i \quad (\text{for } i \in \mathbb{N})$$

gives a **P**-time embedding of $\mathbb{Z}^\omega$ into M_n.

Let $G = \langle a \rangle \; wr \; \langle t \rangle$. A map $e_i \to t^{-i}at^i$, $\quad i \in \mathbb{N}$ gives a **P**-time embedding of $\mathbb{Z}^\omega$ into G.

More examples

Thompson group

The subset sum problem for the Thompson's group

$$F = \langle a, b \mid [ab^{-1}, a^{-1}ba] = 1,\ [ab^{-1}, a^{-2}ba^{2}] = 1\rangle$$

is **NP**-complete.

Proof. The wreath product $\mathbb{Z} \wr \mathbb{Z}$ can be embedded into F.

Baumslag's group *GB*

The subset sum problem for Baumslag's group

$$GB = \langle a, s, t \mid [a, a^{t}] = 1,\ [s, t] = 1,\ a^{s} = aa^{t}\rangle$$

is **NP**-complete.

Thompson group

The subset sum problem for the Thompson's group

$$F = \langle a, b \mid [ab^{-1}, a^{-1}ba] = 1,\ [ab^{-1}, a^{-2}ba^2] = 1 \rangle$$

is **NP**-complete.

Proof. The wreath product $\mathbb{Z} \wr \mathbb{Z}$ can be embedded into F.

Baumslag's group GB

The subset sum problem for Baumslag's group

$$GB = \langle a, s, t \mid [a, a^t] = 1,\ [s, t] = 1,\ a^s = aa^t \rangle$$

is **NP**-complete.

$BS(1,p)$

The subset sum problem for Baumslag-Solitar metabelian group

$$BS(1,p) = \langle a, t \mid t^{-1}at = a^p \rangle$$

is **NP**-complete.

Proof. We showed earlier that $\mathbf{SSP}(\mathbb{Z}, X)$ is **NP**-complete for a generating set $X = \{x_n = 2^n \mid n \in \mathbb{N}\}$. The map

$$x_n \to t^{-n}at^n$$

P-time computable embedding $\phi : \mathbb{Z} \to BS(1,2)$ because $t^{-n}at^n = a^{2^n}$.

Theorem

Let G be a finitely generated virtually nilpotent group. Then **SSP**(G) and **BSMP**(G), as well as their search and optimization variations, are in **P**.

The proof is based on the fact that finitely generated virtually nilpotent groups have polynomial growth.

Theorem

Let G be a hyperbolic group then all the problems $\mathbf{SSP}(G), \mathbf{KP}(G), \mathbf{BSMP}(G)$, as well as their search and optimization versions are in $\mathbf{P}$.