A Distributed Algorithm for Random Convex Programming

L. Carlone[†], V. Srivastava[‡], F. Bullo[‡], G.C. Calafiore[†]

† Dipartimento di Automatica e Informatica
Politecnico di Torino
‡ Center for Control, Dynamical Systems and Computation
University of California Santa Barbara

October 13, 2011 — NetGCoop

Outline

Outline

Random Convex Programs

- Consider the following d-dimensional uncertain convex optimization problem:

$$\min_{x \in X} \; a^\top x$$
$$\text{subject to:} \quad f(x, \delta) \leq 0$$

  - where $\delta \in \Delta \subset \mathbb{R}^\ell$ is a vector of random parameters, $x \in X \subset \mathbb{R}^d$ is the optimization variable, and $f(x, \delta)$ is convex in x.

- Robust optimization would prescribe to solve the problem over all possible values of the uncertain parameters, i.e.

$$\min_{x \in X} \; a^\top x$$
$$\text{subject to:} \quad \max_{\delta \in \Delta} f(x, \delta) \; \leq 0$$

- The robust formulation
  - may be difficult to solve
  - requires the knowledge of the set Δ
  - may provide conservative results.

Random Convex Programs

- In the random convex program (RCP) framework, instead, the optimality is enforced only over N realizations of the random parameters:

$$\min_{x \in X} \; a^\top x$$
$$\text{subject to:} \quad f(x, \delta^{[j]}) \leq 0, \;\; j = 1, \ldots, N$$

  - the N realizations of the uncertainty are called scenario
  - the corresponding optimal solution x^* is called a scenario solution.

- Then, analytic results allow to assess the violation probability, i.e., the probability that x^* is no longer optimal under a new realization of the uncertainty.

- The RCP formulation
  - is computationally tractable (convex program)
  - does not depend on the distribution of δ nor on Δ
  - x^* is optimal, with high probability, on a new scenario

Random Convex Programs

Example: Estimation

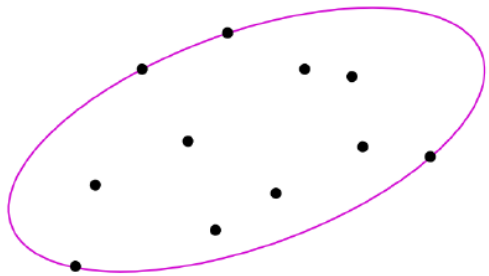
- m sensors can measure a random variable y
- We want to estimate the smallest enclosing ellipsoid (c, M) that contains the N available measurements
- According to the RCP framework, such ellipsoid, with high probability, will contain a future realization of y

- RCP formulation:

$$\min_{c,M} \quad \log\det(M^{-1}) \quad \text{subject to:}$$

$$(y^{[j]} - c)^\top M (y^{[j]} - c) - 1 \leq 0$$

$$j = 1, \ldots, N$$



Random Convex Programs

Example: Linear Classification

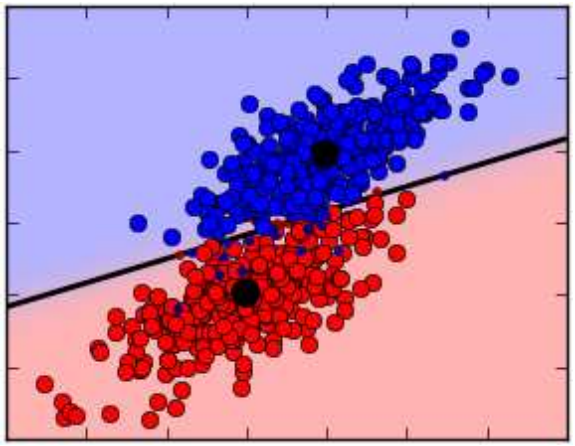
- We have *m* sensors deployed in a region and they:
  - can acquire uncertain measurements of their position a_k (e.g., using GPS)
  - have a unique label b_k that indicates if the region in which the sensor is deployed is safe of not (toxicity, fire, pollution, etc.)
- The objective is to compute a separating hyperplane (θ, ρ) that divides the safe region from the unsafe one

- RCP formulation:

$$\min_{\theta,\lambda,\rho} \quad r(\theta) + \lambda \qquad \text{subject to:}$$

$$I(b_k(\theta^T a_k^{[j]} + \rho)) - \lambda \leq 0,$$

$$j = 1, \ldots, N, \quad k = 1, \ldots, m$$



Outline

Distributed RCP

Motivations

- decentralized nature of the information: in several applications the information is intrinsically distributed among different nodes
  - sensors and actuators networks
- memory and computational limitations: a single computation unit may not be able to solve the overall RCP due to technological limitations
  - parallel computing

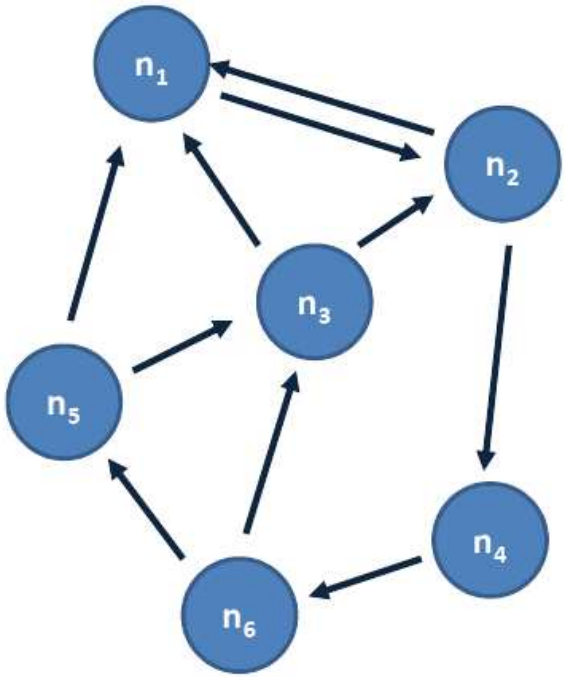
- In our setup the information which is distributed among the nodes is encoded in problem constraints:
  - each node in a network knows a subset of problem constraints and nodes have to interact to solve the overall RCP

[L. Carlone, V. Srivastava, F. Bullo, and G. Calafiore, “A distributed algorithm for random convex programming”, in Proc. of the Int. Conf. on NETwork Games, COntrol and OPtimization (NETGCOOP), pp. 1-7, 2011.]

Distributed RCP

- Denoting with C the constraints set, the global problem is:

$$P[C] \quad \min_{x \in X} \; a^\top x \quad \text{subject to:}$$
$$f(x, \delta^{[j]}) \leq 0, \quad j \in C$$



- In a distributed setup the problem becomes:

$$P[C] \quad \min_{x \in \Omega} \; a^\top x \quad \text{subject to:}$$
$$f_j(x) \leq 0, \quad j \in \cup_{i=i}^{m} C_i$$

$$\begin{cases} \text{where node } i \text{ knows the local set of constraints } C_i \\ \text{can receive information from incoming neighbors } \mathcal{N}_{\text{in}}(i) \\ \text{can transmit information to outgoing neighbors } \mathcal{N}_{\text{out}}(i) \end{cases}$$

Distributed RCP

A Constraints Consensus approach

Definition 1 (Active Constraints)

The active constraints set $ac(C) \subseteq C$ is the set of constraints that are tight at the optimal solution x^*, that is, $ac(C) = \{j \in C : f(x^*, \delta^{[j]}) = 0\}$.

Assumption 1 (Graph connectivity)

The communication digraph is strongly connected, that is, the digraph contains a directed path from each vertex to another.

Assumption 2 (Constraints in general position)

The constraints in the random convex program $P[C]$ are in general position almost surely, i.e., no more than d_{comb} constraints intersect at any point in the domain X ($d_{comb} = d$ for feasible problems, or $d_{comb} = d + 1$ in the possibly infeasible case).

Distributed RCP

A Constraints Consensus approach

- We propose a distributed algorithm for solving random convex programs, named active constraints consensus

- The basic intuitions behind the algorithm are the following:
  - The active constraints set contains all the constraints that are necessary to describe the problem, i.e.,
    - the following problems admit the same solution:

$$P[C]: \min_{x \in X} \; a^\top x \quad \text{s.t.:} \qquad P[\mathrm{ac}(C)]: \min_{x \in X} \; a^\top x \quad \text{s.t.:}$$
$$f_j(x) \leq 0, \quad j \in C \qquad\qquad f_j(x) \leq 0, \quad j \in \mathrm{ac}(C)$$

  - Since constraints are in general position the cardinality of the active constraints set cannot be larger than d_{comb}

Distributed RCP

Active Constraint Consensus

- A generic node i at time t has knowledge of the local set C_i and can store a small candidate set $A_i(t)$.
- Initialization: each node initializes the candidate set to $A_i(0) = \mathtt{ac}(C_i)$, by solving the local convex program $P[C_i]$.
- Iterations: at each iteration t of the algorithm, node i receives the candidate sets from the incoming neighbors and updates its candidate set with the following rule: $A_i(t+1) = \mathtt{ac}\big(A_i(t) \cup \big(\cup_{j \in \mathcal{N}_{\mathsf{in}}(i)} A_j(t)\big) \cup C_i\big)$.
- Halting condition: if the local candidate set does not change for $2\mathtt{diam}(G) + 1$ iterations, the algorithm has reached convergence.

Distributed RCP

Active Constraint Consensus

Proposition 1

Under the assumptions considered so far, the Active Constraint Consensus (ACC) algorithms satisfies the following statements:

1. *At each iteration of the ACC algorithm, each node transmits at most d_{comb} constraints to the outgoing neighbors;*
2. *The local optimal objective is monotonically non-decreasing in t and converges in a finite number of iterations, say T, to the global optimal value $J^*(C)$;*
3. *At iteration T, the local solution at a generic node i is the same as the global solution (scenario solution);*
4. *For each node i the local candidate set $A_i(T)$ at time T coincides with the global active set $ac(C)$.*

Distributed RCP

Active Constraints Consensus: Related Approaches

- The Active Constraints Consensus is similar in spirit to the Constraints Consensus (CC) algorithm:
  - [1] G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: Theory and applications. IEEE Transactions on Automatic Control, 56(10):2247-2261, 2011.
- A similar approach has also been proposed in the context of distributed linear programming:
  - [2] M.Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer. A distributed simplex algorithm for degenerate linear programs and multi-agent assignment. Automatica, 2012. To appear.

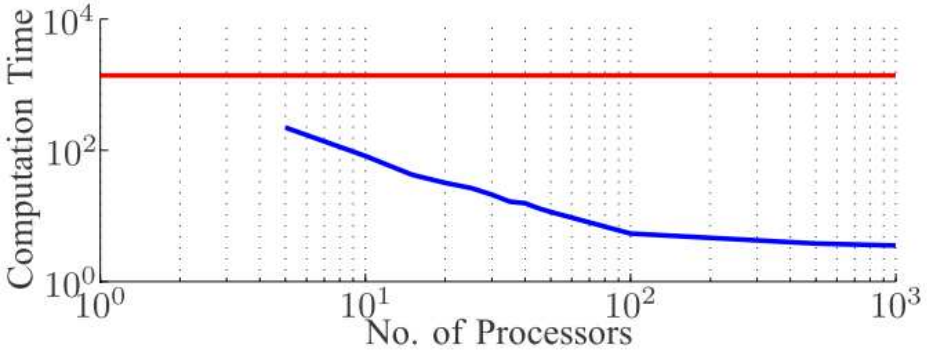
Outline

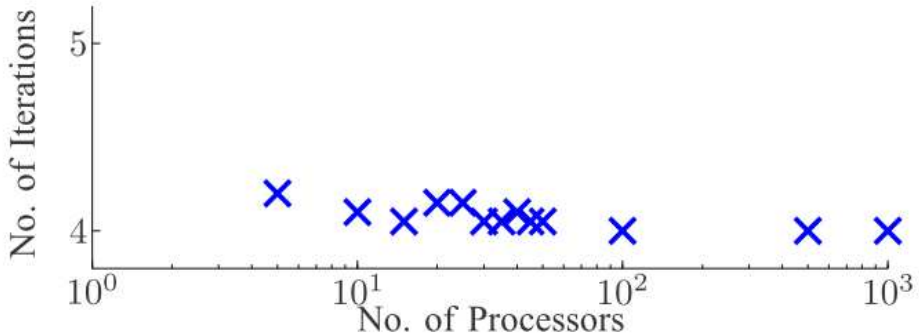
Numerical Examples

Example: Distributed Estimation

- *m* sensors can measure a random variable *y*
- Smallest enclosing ellipsoid
- We consider $y \in \mathbb{R}^2$, distributed according to a mixture distribution
- The overall number of measurements is $N = 200000$

- Parallel Computation (complete communication graph)





Numerical Examples

Example: Distributed Estimation

- Distributed Computation (other graph topologies)

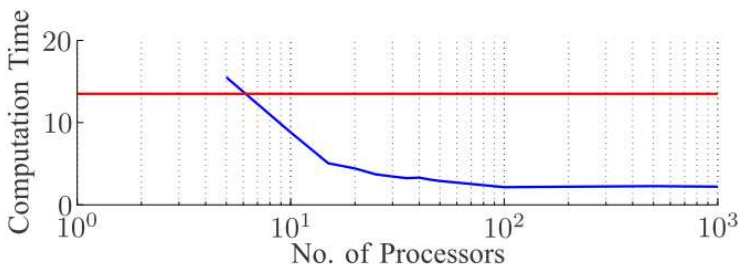
	No. nodes	Diameter	Iterations	Active Constraints
Geometric random graph	10	1	4.1	4.6
	50	2	7.1	
	100	3	9.6	
	500	4	15.5	
Chain graph	10	10	30.7	4.6
	50	50	160	
	100	100	319.7	
	500	500	1628.6	

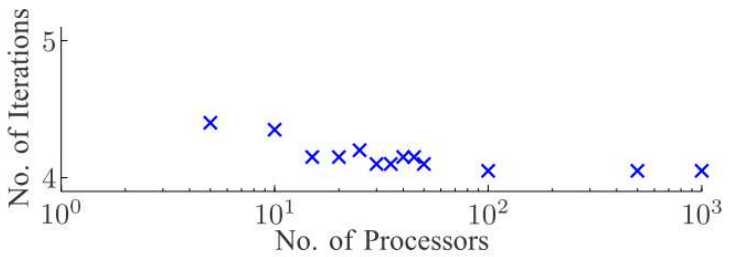
Numerical Examples

Example: Distributed Classification

- m sensors deployed in a region
  - can acquire uncertain measurements of their position a_k
  - have a unique label b_k
- Separating hyperplane
- $N = 200000$ measurements from sensors with different labels

- Parallel Computation (complete communication graph)





Outline

Conclusion

- We investigated the case in which the constraints of a random convex program are distributed among nodes in a network
- We presented a novel algorithm, named active constraints consensus (ACC), that allows to solve a random convex program in a distributed fashion
- In the ACC algorithm nodes exchange small subsets of constraints and can
  - compute the global solution in a finite number of iterations,
  - with sustainable communication effort
- Since a scenario in the RCP framework is a convex program, the algorithm can be easily extended to any convex program
- The ACC algorithm also offers a powerful approach to the parallel computation of the scenario solution, significantly improving the computation time.

Thanks for your attention!