Distributed Estimation via Dual Decomposition

Sikandar Samar
Integrated Data Systems Department
Siemens Corporate Research
Princeton, NJ 08540
sikandar.samar@siemens.com

Stephen Boyd
Information Systems Laboratory
Stanford University
Stanford, CA 94305
boyd@stanford.edu

Dimitry Gorinevsky
Information Systems Laboratory
Stanford University
Stanford, CA 94305
gorin@stanford.edu

Abstract— **The focus of this paper is to develop a framework for distributed estimation via convex optimization. We deal with a network of complex sensor subsystems with local estimation and signal processing. More specifically, the sensor subsystems locally solve a maximum likelihood (or maximum a posteriori probability) estimation problem by maximizing a (strictly) concave log-likelihood function subject to convex constraints. These local implementations are not revealed outside the subsystem. The subsystems interact with one another via convex coupling constraints. We discuss a distributed estimation scheme to fuse the local subsystem estimates into a globally optimal estimate that satisfies the coupling constraints. The approach uses dual decomposition techniques in combination with the subgradient method to develop a simple distributed estimation algorithm. Many existing methods of data fusion are suboptimal, *i.e.*, they do not maximize the log-likelihood exactly but rather 'fuse' partial results from many processors. For linear gaussian formulation, least mean square (LMS) consensus provides optimal (maximum likelihood) solution. The main contribution of this work is to provide a new approach for data fusion which is based on distributed convex optimization. It applies to a class of problems, described by concave log-likelihood functions, which is much broader than the LMS consensus setup.**

I. Introduction

This paper is about data fusion, which is estimation with the help of distributed sensors and distributed processors. We consider a system consisting of a collection of sensor subsystems, each receiving noisy measurements pertaining to its unknown parameters. The sensor subsystems have their own local estimation routines, which we assume involve maximization of a concave log-likelihood function subject to convex constraints. The subsystems interact with one another via coupling (consistency) constraints. The goal is to fuse the individual subsystem estimates to obtain a globally maximal likelihood estimate, subject to the consistency constraints.

There is a large relevant body of prior literature on the general theory of distributed algorithms [Lyn96], [Tel94], distributed and parallel computation [BT97], and distributed optimization [Sim91], [HK00]. The distributed estimation framework lends itself to the use of *decomposition methods*. Decomposition is a standard method used to solve a large-scale problem by breaking it up into smaller subproblems, and solving the subproblems independently (locally) [GSV01]. The challenge is to coordinate the solution of the subproblems to achieve globally optimal and consistent estimates. Decomposition methods have a long history in optimization, going back to the Dantzig-Wolfe decomposition [DW60] and Benders decomposition [Ben62]. A more recent reference on decomposition methods is [Ber99]. For decomposition applications applied to networking problems see [KMT97], [CLCD07]. These methods were originally developed to exploit problem structure in order to achieve significant gains in computational efficiency and reductions in computer memory requirements. Decomposition methods support the isolation of the subsystems except through a small interface, which is of great interest in distributed implementation.

We combine the dual decomposition approach with the *subgradient method*, which is a simple algorithm for minimizing a nondifferentiable convex function. Minimizing the dual function using the subgradient method

This material is based upon work supported by the National Science Foundation under grants #0423905 and (through October 2005) #0140700, by the Air Force Office of Scientific Research under grant #F49620-01-1-0365, and by MARCO Focus center for Circuit & System Solutions contract #2003-CT-888.

preserves the distributed architecture of the estimation problem. Subgradient methods were developed in the 1970s by Shor in the Soviet Union. The main reference is his book [Sho85]. Other useful books on this topic include [NY83], [Akg84], [Pol87], [CZ97], [Sho98], [Ber99], [Nes04], [Rus06]. Some interesting recent research papers on subgradient methods are [NB01] and [Nes05].

This paper is organized as follows: Section II describes the general framework for the distributed estimation problem. In Section III we give a simple conceptual example to explain the general framework. Section IV describes the solution approach based on dual decomposition and subgradient method. A large scale target tracking example is presented in Section V to illustrate the proposed approach. Some concluding remarks are given in Section VI.

II. Framework for distributed estimation

We describe a general framework for distributed estimation where we have K subsystems. The subsystems are complex groups of sensors with local estimation and signal processing. We let $x_i \in \mathbf{R}^{n_i}$ and $y_i \in \mathbf{R}^{p_i}$ denote the unknown private and public variables of subsystem i. Each subsystem has a local (strictly) concave log-likelihood function $l_i : \mathbf{R}^{n_i} \times \mathbf{R}^{p_i}$. The overall log-likelihood function of the system is given by

$$l(x,y) \quad = \quad l_1(x_1,y_1) + \ldots + l_K(x_K,y_K).$$

If the subsystem variables are a priori known to lie in a constraint set $\mathcal{C}_i \subseteq \mathbf{R}^{n_i} \times \mathbf{R}^{p_i}$, we have local constraints

$$(x_i, y_i) \in \mathcal{C}_i.$$

Here $\mathcal{C}_i$ is a feasible set for subsystem i, presumably described by linear equalities and convex inequalities.

Additionally, we allow interaction between the subsystems. We assume the K subsystems are coupled through constraints that require various subsets of components of the public variables to be equal. These constraints are called *consistency constraints*. In order to describe these constraints we collect all the public variables together into one vector variable $y = (y_1, \ldots, y_K) \in \mathbf{R}^p$, where $p = p_1 + \cdots + p_K$ is the total number of (scalar) public variables. We use the notation $(y)_i$ to denote the ith (scalar) component of y, for $i = 1, \ldots, p$ (in order to distinguish it from y_i, which refers to the portion of y associated with subsystem i).

We suppose there are N consistency constraints, and introduce a vector $z \in \mathbf{R}^N$ that gives the common values of the public variables in each consistency constraint. We can express the constraints as

$$y = Ez,$$

where $E \in \mathbf{R}^{p \times N}$ is the matrix with

$$E_{ij} = \begin{cases} 1 & (y)_i \text{ is in constraint } j \\ 0 & \text{otherwise.} \end{cases}$$

The matrix E specifies the set of consistency constraints for the given subsystem interaction. We will let $E_i \in \mathbf{R}^{p_i \times N}$ denote the partitioning of the rows of E into blocks associated with the different subsystems, so that $y_i = E_i z$.

The maximum likelihood estimation problem in the presence of subsystem interactions is given by

$$\begin{array}{ll} \text{maximize} & l_1(x_1,y_1) + \cdots + l_K(x_K,y_K) \\ \text{subject to} & (x_i,y_i) \in \mathcal{C}_i, \quad i = 1, \ldots, K \\ & y_i = E_i z, \quad i = 1, \ldots, K, \end{array} \tag{1}$$

with variables x_i, y_i, and z. Note that the problem is completely separable in the absence of consistency constraints.

A. Hypergraph Representation

We can represent this distributed estimation framework as a hypergraph. The nodes in the hypergraph are associated with individual subsystems which have local log-likelihood functions and local constraints. The hyperedges (or nets) are associated with consistency constraints. A link is a hyperedge between two nodes and corresponds to a constraint between the two subsystems represented by the nodes. z is a vector of net variables that specifies the common values of the public variables on the N nets. The matrix E specifies the complete netlist, or set of hyperedges, for all the subsystem interactions. The matrix E_i in (1) is a 0-1 matrix that maps the vector of net variables into the public variables of subsystem i.

B. Applications

Estimation problems with an inherently distributed architecture arise naturally in many applications in diagnostics, networks, image processing, and target tracking. In distributed diagnostics, we have a collection of

sensor subsystems estimating damage at various points on a particular surface. The local constraints in a damage estimation problem may represent monotonicity of damage parameters, *i.e.*, damage can only get worse. The consistency constraints can be used to specify the requirement that damage estimates at a (common) point obtained by different subsystems should agree. In some image processing problems, pixels are only coupled to some of their neighbors. In this case, any strip with a width exceeding the interaction distance between pixels, and which disconnects the image plane, represents the coupling constraints between partitioned image problems.

III. Simple class example: Three subsystems

We describe the general framework of Section II by a simple conceptual example. Figure 1 shows a system with three subsystems labeled 1, 2, and 3. Subsystem 1 has private variables $x_1 \in \mathbf{R}^{n_1}$ and a public variable $y_1 \in \mathbf{R}$. Subsystem 2 has private variables $x_2 \in \mathbf{R}^{n_2}$ and a public variable $y_2 \in \mathbf{R}^2$. Subsystem 3 has private variables $x_3 \in \mathbf{R}^{n_3}$ and a public variable $y_3 \in \mathbf{R}$. This system has a chain structure with two edges: the first edge corresponds to the consistency constraint $y_1 = y_2(1)$, and the second edge corresponds to the consistency constraint $y_2(2) = y_3$.

In order to describe these constraints in the desired form, we collect the three public variables into one vector variable $y = (y_1, y_2, y_3) \in \mathbf{R}^4$. The two constraints in terms of the components of y are $(y)_1 = (y)_2$ and $(y)_3 = (y)_4$. We introduce a vector $z = (z_1, z_2) \in \mathbf{R}^2$ that gives the common values of the public variables on each link. The rows of matrix $E \in \mathbf{R}^{4\times 2}$ that specifies the two links can be partitioned into three blocks associated with the three subsystems where

$$E_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

The maximum likelihood estimation problem of this system with the consistency constraints is

$$\begin{array}{ll} \text{maximize} & l_1(x_1, y_1) + l_2(x_2, y_2) + l_3(x_3, y_3) \\ \text{subject to} & (x_i, y_i) \in \mathcal{C}_i, \quad i = 1, 2, 3 \\ & y_i = E_i z, \quad i = 1, 2, 3. \end{array}$$

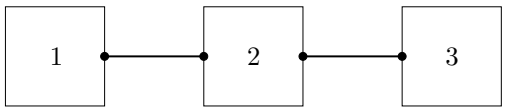

Fig. 1. Chain structure, with three subsystems and two coupling constraints.

IV. Dual decomposition

We use dual decomposition to solve the maximum likelihood estimation problem (1) in a distributed setting. The first step is to form the partial Lagrangian, by introducing Lagrange multipliers only for the coupling constraint

$$\begin{aligned} L(x, y, z, \nu) &= \sum_{i=1}^{K} l_i(x_i, y_i) - \nu^T (y - Ez) \\ &= \sum_{i=1}^{K} \left(l_i(x_i, y_i) - \nu_i^T y_i \right) + \nu^T E z, \end{aligned}$$

where $\nu \in \mathbf{R}^p$ is the Lagrange multiplier associated with $y = Ez$, and ν_i is the subvector of ν associated with the ith subsystem. We let q denote the dual function. To find the dual function we first maximize over z, which results in the condition $E^T\nu = 0$ for $q(\nu) < \infty$. This condition states that for each net, the sum of the Lagrange multipliers over the net is zero.

We can now solve the following subproblems independently for each subsystem i give ν

$$\begin{array}{ll} \underset{x_i, y_i}{\text{maximize}} & l_i(x_i, y_i) - \nu_i^T y_i \\ \text{subject to} & (x_i, y_i) \in \mathcal{C}_i. \end{array} \tag{2}$$

We assume that the local log-likelihood functions $l_i(x_i, y_i)$ are strictly concave. The solution (x_i^*, y_i^*) to each subproblem is therefore unique. If we use $q_i(\nu)$ to denote the optimal value of each subproblem, then the dual of the original problem (1) with variable ν is

$$\begin{array}{ll} \underset{\nu}{\text{minimize}} & q(\nu) = \sum_{i=1}^{K} q_i(\nu_i) \\ \text{subject to} & E^T\nu = 0. \end{array} \tag{3}$$

We refer to (3) as the dual decomposition *master problem*. We assume that strong duality holds, *i.e.*, the duality gap reduces to zero. This implies that the primal problem (1) can be equivalently solved by solving the dual problem (3).

A. Solution via subgradient method

In this paper, we use a projected subgradient method to develop a simple decentralized algorithm to solve the dual master problem. The basic subgradient method for an unconstrained problem uses the iteration

$$\nu^{(k+1)} \quad = \quad \nu^{(k)} - \alpha_k g^{(k)},$$

where $\nu^{(k)}$ is the kth iterate, $g^{(k)}$ is any subgradient of $q(\nu)$ at the $\nu^{(k)}$, and $\alpha_k > 0$ is the kth step size. At each iteration of the subgradient method, we take a step in the direction of negative subgradient. A subgradient of q_i at ν_i is simply $-y_i^*$, which is the optimal value of y_i in the subproblem (2). The projected subgradient method is given by

$$\nu^{(k+1)} = P\left(\nu^{(k)} - \alpha_k g^{(k)}\right),$$

where P is the (Euclidean) projection on the feasible set for ν. Since the feasible set in problem (3) is affine, *i.e.*, $\{\nu \mid E^T\nu = 0\}$ where E^T is fat and full rank, the projection operator is affine, and given by

$$P(z) = z - E(E^TE)^{-1}E^Tz.$$

In this case, we can simplify the subgradient update to

$$\begin{aligned}\nu^{(k+1)} \quad &= \quad \nu^{(k)} - \alpha_k(I - E(E^TE)^{-1}E^T)g^{(k)} \\ &= \quad \nu^{(k)} - \alpha_k P_{\mathcal{N}(E^T)}g^{(k)}.\end{aligned}$$

Thus, we simply project the current subgradient onto the nullspace of E^T, and then update as usual. Note that this update is not the same as the projected subgradient update when the feasible set is not affine, because in this case the projection operator is not affine. The projection onto the feasible set $\{\nu \mid E^T\nu = 0\}$, simply consists of vectors whose sum over each net is zero. This is particularly easy to work out since

$$E^TE = \mathbf{diag}(d_1, \ldots, d_N),$$

where d_i is the degree of net i, *i.e.*, the number of subsystems adjacent to net i. For $u \in \mathbf{R}^p$, $(E^TE)^{-1}E^Tu$ gives the average, over each net, of the entries in the vector u. Thus, $(E^TE)^{-1}E^Tu$ is the vector obtained by replacing each entry of u with its average over its associated net. Finally, projection of u onto the feasible set is obtained by subtracting from each entry the average of other values in the associated net. Dual decomposition, with a subgradient method for the master problem, gives the following algorithm.

given initial price vector ν that satisfies $E^T\nu = 0$ (*e.g.*, $\nu = 0$).

repeat

Optimize subsystems (separately).
Solve subproblems (2) to obtain x_i^*, y_i^*.
Compute average of public variables over each net.
$\hat{z} := (E^TE)^{-1}E^Ty^*$.
Update the dual variables.
$\nu := \nu - \alpha_k(-y^* + E\hat{z})$.

The norm of the vector computer in the last step, *i.e.*, $\|y^* - E\hat{z}\|$, gives a measure of the inconsistency of the current values of the public variables and is called the *consistency constraint residual*.

This algorithm is decentralized: Each subsystem has its own private copy of the public variables on the nets it is adjacent to, as well as an associated dual variable. Using these dual variables, the subsystems first optimize independently by solving a (local) concave maximization problem. At the second step, the nets, also acting independently of each other, update the value of the net variable using the optimal values of the public variables of the subsystems adjacent to that net. The dual variables are then updated, in a way that brings the local copies of public variables into consistency (and therefore also optimality).

B. Economic Interpretation of Dual Decomposition

Dual decomposition has an interesting economic interpretation. Consider for example the system shown in Figure 1. We can imagine the three subsystems as separate economic units, each with its own utility function, private variables and public variables. We can think of y_1 as the amount of some resources generated by the first unit, and $y_2(1)$ as the amount of some resources consumed by the second unit. Then, the consistency constraint $y_1 = y_2(1)$ means that supply of the first unit equals demand of the second unit. Similarly, we can think of the constraint $y_2(2) = y_3$ as a supply demand equilibrium between the second and third units. We interpret the dual variables ν as prices for the resources. At each iteration of the algorithm, we set the prices for the resources. Then, each unit operates independently in such a way that its utility for the given price is maximized. The dual decomposition algorithm adjusts the prices in order to bring the supply into consistency with

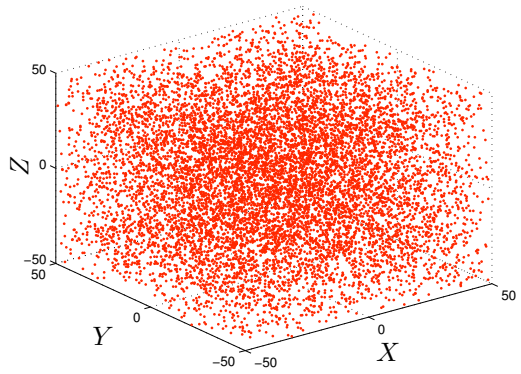

Fig. 2. Unknown Target locations in $\mathbf{R}^3$

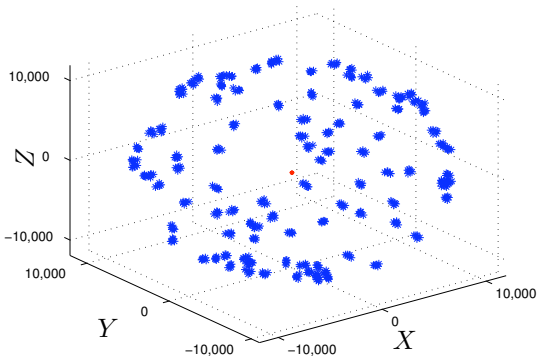

Fig. 3. Subsystems on the surface of sphere of radius 12000

the demand at each link. This is achieved by increasing the prices in over-subscribed resources and reducing the prices on under-subscribed resources. In economics, this is also called the *price adjustment algorithm*.

V. Numerical example: Target Tracking

In this section, we apply the distributed estimation framework to a large-scale target tracking example. We consider $10,000$ unknown target locations in $\mathbf{R}^3$. The goal is to estimate the target locations using 100 subsystems (navigation satellites), each with 6 sensors. This means that a subsystem has 6 measurements for every target that it estimates. Since each target location is specified by three co-ordinates, we roughly have a $2:1$ measurement redundancy ratio.

Figure 2 shows $10,000$ target locations generated from a uniform distribution on the interval $[-50, 50]$. Since typical navigation satellites are about $12,000$ miles from the earth, the subsystems are randomly placed on the surface of a sphere of radius 12000; see Figure 3. Due to the close proximity of the targets to the origin and the distant location of the subsystems, the linearization around zero is (say) nearly exact.

In practice, the subsystems will only estimate the targets within their range. The private variables in this example correspond to those target locations that are only estimated by one subsystem. We consider a hypothetical scenario where each subsystem estimates a total of 110 targets in $\mathbf{R}^3$, *i.e.*, $n_i + p_i = 330$ parameters. Thus, there is an overlap of about 10% unknown target locations, *i.e.*, roughly $1,000$ targets are estimated by multiple subsystems. The consistency constraints require that estimates of a particular target location computed by multiple subsystems should agree. We collect all the public and private variables of each subsystem into one vector $\theta_i = (x_i, y_i) \in \mathbf{R}^{330}$. With the linear measurement model assumption, the maximum likelihood estimation problem is

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{100} \|b_i - A_i\theta_i\|_2^2 \\ \text{subject to} & l_i \le \theta_i \le u_i, \quad i = 1, \ldots, 100 \\ & y_i = E_i z, \end{array}$$

where we minimize the negative log-likelihood function. Each row of the matrix $A_i \in \mathbf{R}^{660\times 330}$ is a unit vector from zero to a sensor location of subsystem i. The measurement vector $b_i \in \mathbf{R}^{660}$ is obtained by adding a 5% uniform noise to $A_i\theta_i$. The subsystems compute their estimates of the unknown targets in the presence of (local) box constraints $[l_i, u_i]$.

We solve the problem using the dual decomposition approach of the previous section. A subgradient method with a diminishing step size $\alpha_k = 0.02/\sqrt{k}$ was used for this particular example. The algorithm was run for 50 iterations. For the diminishing step size rule, the subgradient method is guaranteed to converge to the optimal value, *i.e.*, $\lim_{k\to\infty} q(\nu^k) = q^*$. The strong duality implies $q^* = p^*$, where $p^* \approx 7.7$ is the optimal primal objective value. Figure 4 shows the duality gap in the first 50 iterations. The flat regions in the plot indicate that the dual function value actually decreased during those iterations. When using the subgradient method, it

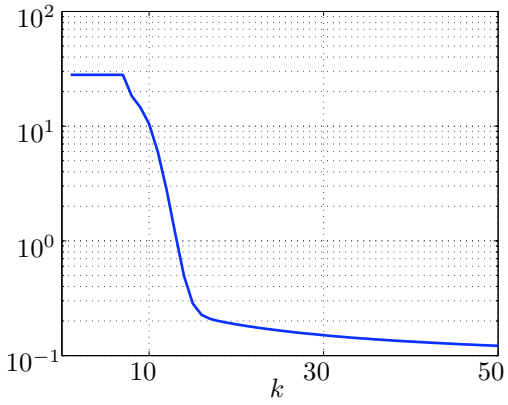

Fig. 4. Duality gap $p^* - q^*$, versus iteration number k.

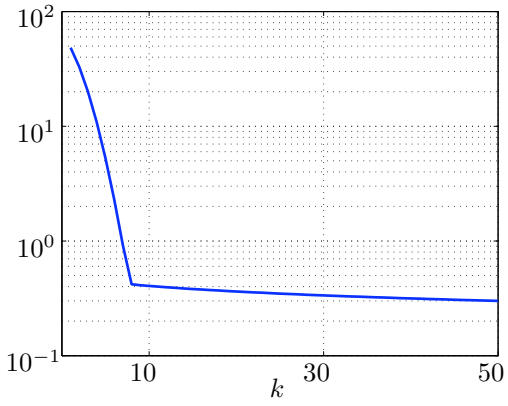

Fig. 5. Norm of the consistency constraint residual $\|y - Ez\|$, versus iteration number k.

is therefore customary to keep track of the best dual function value. The duality gap approaches zero with increasing number of iterations. Figure 5 shows the norm of the consistency constraint residual versus the iteration number.

VI. Conclusion

In this paper, we present a distributed estimation approach using convex optimization algorithms. The presented approach extends the current distributed average consensus problems that limit the subsystems to solving linear unconstrained Gaussian estimation problems. We use a dual decomposition approach and subgradient method to provide a tractable solution for large-scale convex distributed estimation problems. The approach is illustrated through an example from target tracking.

References

[Akg84] M. Akgül. *Topics in Relaxation and Ellipsoidal Methods*, volume 97 of *Research Notes in Mathematics*. Pitman, 1984.

[Ben62] J. F. Benders. Partioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[Ber99] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[BT97] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[CLCD07] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, January 2007. To appear.

[CZ97] Y. Censor and S. Zenios. *Parallel Optimization*. Oxford University Press, 1997.

[DW60] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[GSV01] J. Gondzio, R. Sarkissian, and J.-Ph. Vial. Parallel implementation of a central decomposition method for solving large-scale planning problems. *Comput. Optim. Appl.*, 19(1):5–29, 2001.

[HK00] B. Hendrickson and T. G. Kolda. Graph partitioning models for parallel computing. *Parallel Computing*, 26(12):1519–1534, 2000.

[KMT97] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1997.

[Lyn96] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996.

[NB01] A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 12:109–138, 2001.

[Nes04] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[Nes05] Y. Nesterov. Primal-dual subgradient methods for convex problems. CORE Discussion Paper #2005/67. Available at `www.core.ucl.ac.be/services/psfiles/dp05/dp2005_67.pdf`, 2005.

[NY83] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[Pol87] B. T. Polyak. *Introduction to Optimization*. Optimization Software Inc., New York, 1987.

[Rus06] A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.

[Sho85] N. Z. Shor. *Minimization Methods for Non-differentiable Functions*. Springer Series in Computational Mathematics. Springer, 1985.

[Sho98] N. Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*. Nonconvex Optimization and its Applications. Kluwer Academics Publishers, 1998.

[Sim91] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2:135–148, 1991.

[Tel94] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, second edition, 1994.