A Proximal Bundle Method for Nonsmooth DC Optimization

Kaisa Joki[1]

kaisa.joki@utu.fi

Adil Bagirov[2], Napsu Karmitsa[1] and Marko M. Mäkelä[1]

[1]University of Turku

[2]Federation University Australia

14.11.2014

OSE Seminar 2014

Outline

Nonsmooth DC Optimization

- Functions can be presented as a difference of two convex functions and such functions are called DC functions
- Functions need not to be differentiable
- The general problem is that nonsmooth functions are typically not differentiable at their minimizers
- When the gradient does not exist at every point, we cannot utilize the classical theory of optimization or smooth gradient based methods

DC Optimization

- Any twice continuously differentiable function can be presented as a DC function
- Any continuous function can be approximated by the sequence of DC functions
- Many optimization problems can be expressed into the form of a DC program such as
  - Production-transportation planning
  - Location planning
  - Engineering design
  - Cluster analysis
  - Multiobjective programming

Problem

Nonsmooth DC problem

We consider an unconstrained minimization problem of the form

$$\begin{cases} \min & f(\boldsymbol{x}) \\ \mathrm{s.\,t.} & \boldsymbol{x} \in \mathbb{R}^n, \end{cases}$$

where

- Objective function $f : \mathbb{R}^n \to \mathbb{R}$ is
  - assumed to be a DC function
  - not required to have continuous derivatives

DC Functions

Definition 1

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called a *DC function* if it can be written in the form

$$f(\boldsymbol{x}) = f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}),$$

where f_1 and f_2 are convex functions on $\mathbb{R}^n$.

- Functions f_1 and f_2 are called DC components of f and in what follows they are assumed be finite on $\mathbb{R}^n$
- If f is nonsmooth then at least one of the functions f_1 and f_2 is nonsmooth
- DC functions are locally Lipschitz continuous and usually also nonconvex

Turun yliopisto
University of Turku

Convex Analysis

Next we consider the convex DC components $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2$.

Definition 2

The *subdifferential* of f_i at $\boldsymbol{x} \in \mathbb{R}^n$ is a set

$$\partial f_i(\boldsymbol{x}) = \left\{ \boldsymbol{\xi}_i \in \mathbb{R}^n \mid f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x}) + \boldsymbol{\xi}_i^T(\boldsymbol{y} - \boldsymbol{x}) \text{ for all } \boldsymbol{y} \in \mathbb{R}^n \right\}.$$

Each vector $\boldsymbol{\xi}_i \in \partial f_i(\boldsymbol{x})$ is called a *subgradient* of f_i at $\boldsymbol{x}$.

Definition 3

Let $\varepsilon \geq 0$, the ε-*subdifferential* of f_i at $\boldsymbol{x} \in \mathbb{R}^n$ is a set

$$\partial_\varepsilon f_i(\boldsymbol{x}) = \left\{ \boldsymbol{\xi}_i \in \mathbb{R}^n \mid f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x}) + \boldsymbol{\xi}_i^T(\boldsymbol{y} - \boldsymbol{x}) - \varepsilon \text{ for all } \boldsymbol{y} \in \mathbb{R}^n \right\}.$$

Each vector $\boldsymbol{\xi}_i \in \partial_\varepsilon f_i(\boldsymbol{x})$ is called an ε-*subgradient* of f_i at $\boldsymbol{x}$.

Necessary Optimality Condition for a DC Function

Theorem 4

(Toland, 1979) *If $\boldsymbol{x}^* \in \mathbb{R}^n$ is a local minimizer of $f = f_1 - f_2$, then*

$$\partial f_2(\boldsymbol{x}^*) \subset \partial f_1(\boldsymbol{x}^*).$$

Definition 5

Let $\varepsilon \geq 0$, a point $\boldsymbol{x}^* \in \mathbb{R}^n$ is called an ε-*critial point*, if it satisfies the condition

$$\partial_\varepsilon f_2(\boldsymbol{x}^*) \cap \partial_\varepsilon f_1(\boldsymbol{x}^*) \neq \emptyset.$$

If $\varepsilon = 0$, then $\boldsymbol{x}^*$ is said to be a *critical point*.

- Solution candidates of our bundle method PBDC are ε-critial points

Turun yliopisto
University of Turku

About the New Cutting Plane Model

- Used to determine a search direction in our bundle algorithm
- Utilizes explicitly the DC decomposition of the objective function f
- The main idea in model construction:
  - Approximate the subdifferentials of both DC components f_i with a *bundle*
  - Two separate bundles which consist of subgradients from the previous iterations
  - Use subgradient information to construct separately an approximation for each DC component f_i
  - Combine the separate approximations to obtain a piecewise linear cutting plane model for the original objective function f

Turun yliopisto
University of Turku

Bundles for DC Components

- Assumption: At each point $\boldsymbol{x} \in \mathbb{R}^n$ we can evaluate the values of DC components $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ as well as arbitrary subgradients $\boldsymbol{\xi}_1 \in \partial f_1(\boldsymbol{x})$ and $\boldsymbol{\xi}_2 \in \partial f_2(\boldsymbol{x})$
- At the current iteration point $\boldsymbol{x}_k$ our *bundle* for f_i is denoted by

$$\mathcal{B}_i^k = \big\{ (\boldsymbol{y}_j, f_i(\boldsymbol{y}_j), \boldsymbol{\xi}_{i,j}) \,|\, j \in J_i^k \big\},$$

where

  - the subscript i tells the DC component f_i in question
  - $\boldsymbol{y}_j \in \mathbb{R}^n$ is an auxiliary point
  - $\boldsymbol{\xi}_{i,j} \in \partial f_i(\boldsymbol{y}_j)$ is a subgradient
  - J_i^k is a nonempty set of indices

Turun yliopisto
University of Turku

Approximations for DC Components

- A convex piecewise linear approximation of the convex DC component f_i can be constructed by

$$\hat{f}_i^k(\boldsymbol{x}) = \max_{j \in J_i^k} \left\{ f_i(\boldsymbol{x}_k) + (\boldsymbol{\xi}_{i,j})^T (\boldsymbol{x} - \boldsymbol{x}_k) - \alpha_{i,j}^k \right\}$$

  with the *linearization error*

$$\alpha_{i,j}^k = f_i(\boldsymbol{x}_k) - f_i(\boldsymbol{y}_j) - (\boldsymbol{\xi}_{i,j})^T (\boldsymbol{x}_k - \boldsymbol{y}_j) \geq 0 \quad \text{for all } j \in J_i^k$$

- $\hat{f}_i^k(\boldsymbol{x})$ is a convex function and $\hat{f}_i^k(\boldsymbol{x}) \leq f_i(\boldsymbol{x})$
- This approximation is the classical cutting plane model used in convex bundle methods (see e.g.: Kiwiel, 1990; Mäkelä, 2002)

Turun yliopisto
University of Turku

New Cutting Plane Model

- The new *cutting plane model* of the objective function f is defined by

$$\hat{f}^k(\boldsymbol{x}) = \hat{f}_1^k(\boldsymbol{x}) - \hat{f}_2^k(\boldsymbol{x})$$

- This model can be rewritten in an equivalent form

$$\hat{f}^k(\boldsymbol{x}_k + \boldsymbol{d}) = f(\boldsymbol{x}_k) + \Delta_1^k(\boldsymbol{d}) + \Delta_2^k(\boldsymbol{d}),$$

where

  - $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}_k$ is the search direction
  - $\Delta_1^k(\boldsymbol{d}) = \max_{j \in J_1^k} \left\{ (\boldsymbol{\xi}_{1,j})^T \boldsymbol{d} - \alpha_{1,j}^k \right\}$
  - $\Delta_2^k(\boldsymbol{d}) = \min_{j \in J_2^k} \left\{ -(\boldsymbol{\xi}_{2,j})^T \boldsymbol{d} + \alpha_{2,j}^k \right\}$

Search Direction

To determine the search direction $\boldsymbol{d}_t^k$ we need to solve globally the nonsmooth nonconvex DC problem

$$\min_{\boldsymbol{d}\in\mathbb{R}^n} \left\{ P^k(\boldsymbol{d}) = \Delta_1^k(\boldsymbol{d}) + \Delta_2^k(\boldsymbol{d}) + \frac{1}{2t}\|\boldsymbol{d}\|^2 \right\} \tag{1}$$

where

- t is the classical proximity parameter used in bundle methods
- $\frac{1}{2t}\|\boldsymbol{d}\|^2$ is a stabilizing term which
  - guarantees the existence of the solution $\boldsymbol{d}_t^k$
  - keeps the approximation local enough

The solution $\boldsymbol{d}_t^k$ is got by using a specific approach (An & Tao, 1997)

Subproblems

- The objective function $P^k(\boldsymbol{d})$ can be written as

$$P^k(\boldsymbol{d}) = \min_{i \in J_2^k} \left\{ P_i^k(\boldsymbol{d}) = \Delta_1^k(\boldsymbol{d}) - (\boldsymbol{\xi}_{2,i})^T \boldsymbol{d} + \alpha_{2,i}^k + \frac{1}{2t}\|\boldsymbol{d}\|^2 \right\}$$

- Hence the problem (1) takes the form

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \min_{i \in J_2^k} \left\{ P_i^k(\boldsymbol{d}) \right\} = \min_{i \in J_2^k} \min_{\boldsymbol{d} \in \mathbb{R}^n} \left\{ P_i^k(\boldsymbol{d}) \right\}$$

- To obtain the solution $\boldsymbol{d}_t^k$ of the problem (1) we first solve separately for each $i \in J_2^k$ the convex subproblem

$$\min_{\boldsymbol{d} \in \mathbb{R}^n} \left\{ P_i^k(\boldsymbol{d}) \right\}$$

Turun yliopisto
University of Turku

Global Solution

- Each subproblem is of the type usually encountered in bundle methods and it can be reformulated as a smooth quadratic problem
- The subproblem minimizer is denoted by $\boldsymbol{d}_t^k(i)$, for $i \in J_2^k$
- The global solution is

$$\boldsymbol{d}_t^k = \boldsymbol{d}_t^k(i^*) \quad \text{where } \; i^* = \arg\min_{i \in J_2^k} \left\{ P_i^k\big(\boldsymbol{d}_t^k(i)\big) \right\}$$

- The value

$$\Delta_1^k(\boldsymbol{d}_t^k) + \Delta_2^k(\boldsymbol{d}_t^k) \leq 0$$

can be used as a predicted descent of f

Turun yliopisto
University of Turku

Assumptions and Global Parameters

The new bundle algorithm PBDC requires the following

- global parameters:
  - the criticality tolerance $\delta > 0$ and the proximity measure $\varepsilon > 0$
  - the decrease and increase parameters $r \in (0,1)$ and $R > 1$
  - the descent parameter $m \in (0,1)$
- assumptions:

A1 The set $\mathcal{F}_0 = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ is compact

A2 Lipschitz constants $L_1 > 0$ and $L_2 > 0$ of f_1 and f_2 are known (or approximated) on the set $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^n \mid d(\boldsymbol{x}, \mathcal{F}_0) \leq \varepsilon\}$.

The algorithm is based on three different bundle methods (Fuduli et al.: 2004a, 2004b, 2013)

PBDC: Proximal Bundle Algorithm for DC Optimization

BUNDLE ALGORITHM

Step 0. ***(Initialization)*** Select a starting point $\boldsymbol{x}_0 \in \mathbb{R}^n$ and global parameters. Set $k = 0$ and initialize the bundles by setting

$$\mathcal{B}_1^k = \left\{ \Big(\boldsymbol{x}_0, f_1(\boldsymbol{x}_0), \boldsymbol{\xi}_1(\boldsymbol{x}_0) \Big) \right\} \quad \text{and} \quad \mathcal{B}_2^k = \left\{ \Big(\boldsymbol{x}_0, f_2(\boldsymbol{x}_0), \boldsymbol{\xi}_2(\boldsymbol{x}_0) \Big) \right\},$$

where $\boldsymbol{\xi}_1(\boldsymbol{x}_0) \in \partial f_1(\boldsymbol{x}_0)$ and $\boldsymbol{\xi}_2(\boldsymbol{x}_0) \in \partial f_2(\boldsymbol{x}_0)$.

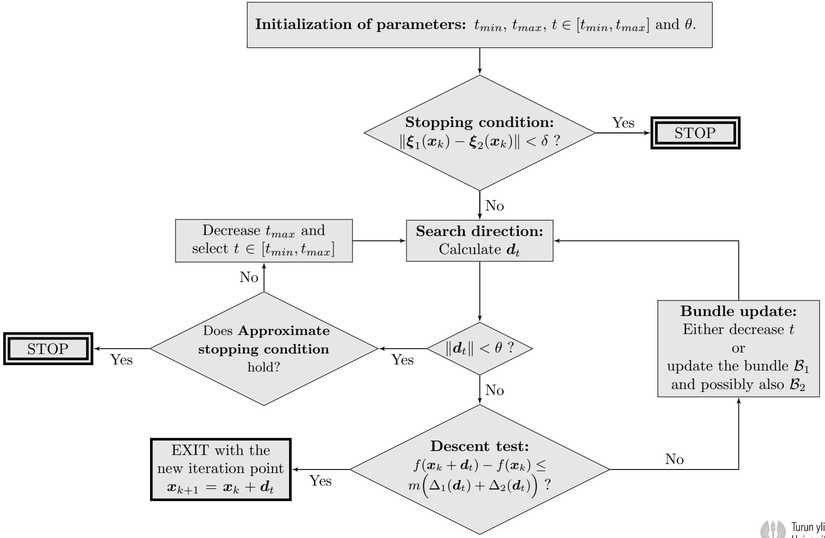
Step 1. ***(Main iteration)*** Execute the 'main iteration'. This either yields a new iteration point $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_t^k$ or stops the algorithm with $\boldsymbol{x}_k$ as the final solution.

Step 2. ***(Bundle update)*** Choose $\mathcal{B}_1^{k+1} \subseteq \mathcal{B}_1^k$ and $\mathcal{B}_2^{k+1} \subseteq \mathcal{B}_2^k$. After this add the element

$$\Big(\boldsymbol{x}_{k+1}, f_1(\boldsymbol{x}_{k+1}), \boldsymbol{\xi}_1(\boldsymbol{x}_{k+1}) \Big) \text{ into } \mathcal{B}_1^{k+1} \text{ and}$$
$$\Big(\boldsymbol{x}_{k+1}, f_2(\boldsymbol{x}_{k+1}), \boldsymbol{\xi}_2(\boldsymbol{x}_{k+1}) \Big) \text{ into } \mathcal{B}_2^{k+1}.$$

Finally set $k = k + 1$ and go back to Step 1.

Main Iteration Algorithm



Approximate Stopping Condition

- For $i = 1, 2$, remove from the bundle $\mathcal{B}_i$ those triplets where the linearization error $\alpha_{i,j} > \varepsilon$
- Calculate values $\boldsymbol{\xi}_1^*$ and $\boldsymbol{\xi}_2^*$ such that

$$\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| = \begin{cases} \min & \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\| \\ \text{s.t.} & \boldsymbol{\xi}_1 \in \operatorname{conv}\{\boldsymbol{\xi}_{1,j} \,|\, j \in J_1\} \text{ and } \boldsymbol{\xi}_2 \in \operatorname{conv}\{\boldsymbol{\xi}_{2,j} \,|\, j \in J_2\}, \end{cases}$$

  where conv denotes the convex hull of a set
- If $\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| < \delta$ then ε-criticality is achieved

Convergence

Theorem 6

The ***'main iteration'*** *terminates after a finite number of steps.*

Theorem 7

For any parameter $\delta > 0$ and $\varepsilon > 0$, the execution of the ***new bundle algorithm*** *stops after a finite number of 'main iterations' at a point $\boldsymbol{x}^*$ satisfying the approximate ε-criticality condition*

$$\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| \leq \delta \quad \text{with } \boldsymbol{\xi}_1^* \in \partial_\varepsilon f_1(\boldsymbol{x}^*) \text{ and } \boldsymbol{\xi}_2^* \in \partial_\varepsilon f_2(\boldsymbol{x}^*).$$

Turun yliopisto
University of Turku

Implementation

- The algorithm PBDC is implemented in double precision Fortran 95
- Subroutines PLQDF1 and PVMM by Lukšan are used to solve the quadratic subproblems and the norm minimization problem (Lukšan: 1984, 2000)
- Results are compared to the proximal bundle algorithm MPBNGC (Mäkelä et al., 1992) and to the truncated codifferential method TCM (Bagirov et al., 2011)
- Tests are performed on an Intel® Core™ i5-2400 CPU (3.10GHz, 3.10GHz)

Parameters Used in PBDC

The parameters of PBDC are tuned as follows:

- the criticality tolerance $\delta = 0.005n$ and the proximity measure $\varepsilon = 0.1$
- the decrease parameter $r = 0.75$, if $n < 10$, and otherwise r is selected to be the first two decimals of $n/(n+5)$
- the increase parameter $R = 10^7$
- the descent parameter $m = 0.2$
- the size of the bundle $\mathcal{B}_1$ is set to $n + 5$
- the size of the bundle $\mathcal{B}_2$ is set to 3
- Lipschitz constants $L_1 = L_2 = 1000$

Nonsmooth Test Problems

Extensions of classical academic minimization problems (Bagirov et al., 2011)

- **Problem 1**

$$
\begin{aligned}
&f_1(\boldsymbol{x}) = \max\{f_1^1(\boldsymbol{x}), f_1^2(\boldsymbol{x}), f_1^3(\boldsymbol{x})\} + f_2^1(\boldsymbol{x}) + f_2^2(\boldsymbol{x}) + f_2^3(\boldsymbol{x}),\\
&f_2(\boldsymbol{x}) = \max\{f_2^1(\boldsymbol{x}) + f_2^2(\boldsymbol{x}), f_2^2(\boldsymbol{x}) + f_2^3(\boldsymbol{x}), f_2^1(\boldsymbol{x}) + f_2^3(\boldsymbol{x})\},\\
&f_1^1(\boldsymbol{x}) = x_1^4 + x_2^2,\ \ f_1^2(\boldsymbol{x}) = (2-x_1)^2 + (2-x_2)^2,\ \ f_1^3(\boldsymbol{x}) = 2e^{-x_1+x_2},\\
&f_2^1(\boldsymbol{x}) = x_1^2 - 2x_1 + x_2^2 - 4x_2 + 4,\ \ f_2^2(\boldsymbol{x}) = 2x_1^2 - 5x_1 + x_2^2 - 2x_2 + 4,\\
&f_2^3(\boldsymbol{x}) = x_1^2 + 2x_2^2 - 4x_2 + 1,\\
&\boldsymbol{x}^* = (1,1), \quad f^* = 2.
\end{aligned}
$$

- **Problem 2** L1 version of Rosenbrock function

$$
\begin{aligned}
&f_1(\boldsymbol{x}) = |x_1 - 1| + 200\max\{0, |x_1| - x_2\}, \quad f_2(\boldsymbol{x}) = 100(|x_1| - x_2)\\
&\boldsymbol{x}^* = (1,1), \quad f^* = 0
\end{aligned}
$$

Turun yliopisto
University of Turku

Nonsmooth Test Problems

- **Problem 3** L1 version of Wood function

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= |x_1 - 1| + 200\max\{0, |x_1| - x_2\} + 180\max\{0, |x_3| - x_4\} \\
&\quad + |x_3 - 1| + 10.1(|x_2 - 1| + |x_4 - 1|) + 4.95|x_2 + x_4 - 2|, \\
f_2(\boldsymbol{x}) &= 100(|x_1| - x_2) + 90(|x_3| - x_4) + 4.95|x_2 - x_4| \\
\boldsymbol{x}^* &= (1,1,1,1), \quad f^* = 0
\end{aligned}
$$

- **Problem 4** DC MaxI

$$
f_1(\boldsymbol{x}) = n\max\left\{|x_i| : \ i = 1,\dots,n\right\}, \quad f_2(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n} |x_i|
$$

$$
|x_i^*| = \alpha \text{ for all } i,\ \alpha \in \mathbb{R}_+, \quad f^* = 0
$$

- **Problem 5**

$$
f_1(\boldsymbol{x}) = x_2 + 0.1(x_1^2 + x_2^2) + 10\max\left\{0, -x_2\right\}, \quad f_2(\boldsymbol{x}) = |x_1| + |x_2|
$$

$$
\boldsymbol{x}^* = (5,0), \quad f^* = -2.5
$$

Turun yliopisto
University of Turku

Numerical Results

		PBDC			MPBNGC
	n	n_f	n_{ξ_1}	n_{ξ_2}	n_f, n_ξ
Problem 1	2	22	17	16	15
Problem 2	2	18	11	11	36
Problem 3	4	23	12	9	61
Problem 4	5	13	6	5	25
	20	25	21	8	130
	100	102	102	30	1433
Problem 5	2	27	19	15	29

TCM	
n_f	n_ξ
246	90
332	107
405	167
235	108
960	451
7064	3471
–	–

- Solvers PBDC and MPBNGC yielded the same global minimizer or best known solution in each test problem
- Solutions obtained by using PBDC were the most accurate ones
- Results for TCM are from the article (Bagirov et al., 2011)

Turun yliopisto
University of Turku

Conclusions

We have developed a new version of the bundle method for nonsmooth DC optimization

- Main requirements are the DC decomposition of f and Lipschitz constants of DC components f_i
- Finds always an ε-critical point after a finite number of steps
- Even though the solution is not necessarily an approximate local minimizer, each local minimizer of f can be found among the set of ε-critical points
- Preliminary computational results seem to be very promising
- Especially the solutions obtained have been nearly always the global minimizers or the best known solutions

References I

[1] L.T.H. An, P.D. Tao: *Solving a Class of Linearly Constrained Indefinite Quadratic Problems by D.C. Algorithms.* J. Global Optim., Vol. 11, 1997, 253–285.

[2] A.M. Bagirov, J. Ugon: *Codifferential method for minimizing nonsmooth DC functions.* J. Global Optim., Vol. 50, 2011, 3–22.

[3] A. Fuduli, M. Gaudioso, G. Giallombardo: *A DC piecewise affine model and bundling technique in nonconvex nonsmooth minimization.* Optimiz. Methods Softw., Vol. 19(1), 2004a, 89–102.

[4] A. Fuduli, M. Gaudioso, G. Giallombardo: *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control.* SIAM J. Optimiz., Vol. 14(3), 2004b, 743–756.

[5] A. Fuduli, M. Gaudioso, E.A. Nurminski: *A splitting bundle approach for non-smooth non-convex minimization.* Optimization, 2013, 1–21. doi: 10.1080/02331934.2013.840625

References II

[6] K.C. Kiwiel: *Proximity control in bundle methods for convex nondifferentiable minimization.* Math. Program., Vol. 46, 1990, 105–122.

[7] L. Lukšan: *Dual method for solving a special problem of quadratic programming as a subproblem at linearly constrained nonlinear minimax approximation.* Kybernetica, Vol. 20, 1984, 445–457.

[8] L. Lukšan, E. Spedicato: *Variable metric methods for unconstrained optimization and nonlinear least squares.* J. Comput. Appl. Math., Vol. 124, 2000, 61–95.

[9] M.M. Mäkelä: *Survey of bundle methods for nonsmooth optimization.* Optimiz. Methods Softw., Vol. 17(1), 2002, 1–29.

[10] M.M. Mäkelä, P. Neittaanmäki: *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., Singapore, 1992.

[11] J.F. Toland: *On subdifferential calculus and duality in nonconvex optimization.* Bull. Soc. Math. France, Mémoire, Vol. 60, 1979, 173–180.

Thank you for your attention!

Turun yliopisto
University of Turku