A TWO-POINT HEURISTIC TO CALCULATE THE STEPSIZE IN SUBGRADIENT METHOD. APPLICATION TO A NETWORK DESIGN PROBLEM.

F. CARRABS*, M. GAUDIOSO†, AND G. MIGLIONICO ‡

Abstract. We introduce a heuristic rule for calculating the stepsize in the subgradient method for unconstrained convex nonsmooth optimization which, unlike the classic approach, is based on retaining some information from previous iteration. The rule is inspired by the well known two-point stepsize by Barzilai and Borwein (BB) [6] for smooth optimization and it coincides with (BB) in case the function to be minimised is convex quadratic.

Under the use of appropriate safeguards we demostrate that the method terminates at a point that satisfies an approximate optimality condition.

The proposed approach is tested in the framework of Lagrangian relaxation for integer linear programming where the Lagrangian dual requires maximization of a concave and nonsmooth (piecewise affine) function. In particular we focus on the relaxation of the Minimum Spanning Tree problem with Conflicting Edge Pairs (MSTC). Comparison with classic subgradient method is presented. The results on some widely used academic test problems are provided too.

Keywords. Convex programming, Subgradient method, Lagrangian relaxation, Minimum Spanning Tree with Conflicting Edge Pairs.

1. Introduction. Subgradient method is the classic tool for dealing with the unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, Lipschitz continuous and not necessarily smooth. We assume that M^*, the set of minima of f is nonempty. It was the first implementable algorithm with proved convergence properties. Detailed presentations are in the seminal books [36] and [32], while a survey of the early stages of growth of numerical nonsmooth optimization are in [26]. Although the birth and the development of the family of Bundle Methods [25], [22] have provided a major improvement of the numerical performance, the subgradient method is still widely used, mainly in tackling the Lagrangian dual in Lagrangian Relaxation [15]. This is possibly due both to its implementation simplicity and to the possibility of using the effective Polyak stepsize [32], [1] when the optimal objective function value is known.

Recent years have seen a renewed interest in theoretical properties of several variants of the subgradient method, specially when some particular structure of the

*Dipartimento di Matematica, Università di Salerno, 84084 Fisciano, Italia. E-mail: fcarrabs@unisa.it

†Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria; ICAR–CNR, 87036 Rende, Italia. E-mail: manlio.gaudioso@unical.it

‡Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, 87036 Rende, Italia. E-mail: g.miglionico@dimes.unical.it

objective function is present. We recall here the contributions in [18], [29], [30], [31], [17] with a rich computational experimentation in [16].

Subgradient–type methods have been extensively investigated also as useful tools for dealing with large scale convex problems arising in the machine learning area. The adaptive gradient method (AdaGrad) [13] belongs to this stream of algorithms. It is applied in cases where the objective function is the sum of many components functions and thus is related to the incremental approach [8].

In this paper we take inspiration from the Barzilai-Borwein method (BB) [6] which is an efficient gradient algorithm for dealing with differentiable optimization. Similarly to the subgradient, it is a non-monotone method and the steplength along the gradient, instead of being provided by a line search, is calculated on the basis of a two-point approximation of the Hessian matrix, taken as a scalar multiple of the identity matrix.

More specifically, letting x_k be the current estimate of the minimum and defining $\delta_k \triangleq x_k - x_{k-1}$ and $\gamma_k \triangleq \nabla f(x_k) - \nabla f(x_{k-1}) = g_k - g_{k-1}$, the approximation B_k of the Hessian $\nabla^2 f(x_k)$ is $B_k = \frac{1}{\alpha_k} I$, where α_k is calculated by solving (in the least squares sense) the secant equation $B_k \delta_k = \gamma_k$. Thus it is

$$\alpha_k = \arg\min_{\alpha} \|\frac{1}{\alpha}\delta_k - \gamma_k\|,$$

and hence we obtain

$$\alpha_k = \frac{\|\delta_k\|^2}{\delta_k^T \gamma_k}, \tag{1.2}$$

which is the BB stepsize adopted, in a classic Newton scheme, to calculate the next iterate

$$x_{k+1} = x_k - \alpha_k g_k.$$

An alternative stepsize can be similarly obtained starting from the approximation of the inverse of the Hessian matrix [6].

Here we introduce yet another possibility to calculate the gradient stepsize. To this aim we define a quadratic model $h_k(d)$ of the difference function $f(x_k+d) - f(x_k)$ by letting

$$h_k(d) = \frac{1}{2} u_k d^\top d + g_k^\top d, \tag{1.3}$$

where u_k is the *proximity* parameter, which is calculated by imposing

$$h_k(d_{k-1}) = f(x_{k-1}) - f(x_k),$$

where $d_{k-1} = x_{k-1} - x_k = -\delta_k$. This leads to the value

$$u_k = \frac{2(f(x_{k-1}) - f(x_k) + g_k^\top \delta_k)}{\|\delta_k\|^2}. \tag{1.4}$$

Calculation of the proximity parameter is similar to the one introduced in [23] in the framework of Bundle methods [19].

By minimizing $h_k(d)$ we obtain

$$d_k \stackrel{\triangle}{=} \arg\min_d h_k(d) = -\frac{1}{u_k} g_k, \tag{1.5}$$

and thus we let

$$x_{k+1} = x_k + d_k = x_k - \frac{1}{u_k} g_k \tag{1.6}$$

If we assume function f to be strictly convex and quadratic, that is $f(x) = \frac{1}{2} x^\top Q x + b^\top x$ for $Q \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, taking into account

$$2\big(f(x_{k-1}) - f(x_k) + g_k^\top \delta_k\big) = \delta_k^\top \gamma_k,$$

we obtain that the stepsize $\frac{1}{u_k}$ in (1.6), with u_k calculated according to (1.4), coincides with Barzilai and Borwein's α_k in (1.2).

Remark 1.1. *Convexity of f guarantees $u_k \geq 0$ since the numerator in (1.4) is the linearization error at x_{k-1} when a linear approximation of f is rooted at x_k. It is bounded away from zero in case f is strongly convex with modulus μ. In this case it is $u_k \geq \mu$.* The main motivation for the use of $\frac{1}{u_k}$ as the stepsize in a nonsmooth framework is that its calculation does not involve difference of gradients, thus it appears more suitable whenever gradient discontinuities can occur. On the other hand, since the underlying model is quadratic, our stepsize is to be considered as a heuristic choice, requiring experimental validation more than a theoretical one.

Thanks to the definition of function $h_k(d)$ and to (1.5), our approach appears as a linearization of the proximal algorithm [33] and, in particular, can be cast in the class of subgradient algorithms with nonlinear projections [7], where, instead of the Euclidean norm, more general distance–like functions are taken in consideration (see [7] also for the relationship of such family of methods with the mirror descent algorithm introduced in [28]).

The rest of the paper is organised as follows. In Section 2 our subgradient method is presented and its termination properties are discussed. In Section 3, as a possible application, we introduce a network design problem known in literature as the Minimum Spanning Tree problem with Conflicting Edge Pairs (MSTC). It is a variant of the classic Minimum Spanning Tree problem where, taking into account a set of conflicting edges, the aim is to determine the cheapest spanning tree with no edge in conflict [37], [34], [9], [10] [11], [35]. We adopt the Lagrangian relaxation scheme discussed in [10], [2], [12] and apply to the resulting Lagrangian dual our subgradient method. The algorithm is tested against several instances from the literature. The results of our algorithm on some academic test problems commonly adopted in convex nonsmooth optimization are presented too. Some conclusions are drawn in Section 4.

2. The subgradient method. The iterative scheme of the classic subgradient method for minimization of a convex and not necessarily differentiable function is analogous to the gradient method, the only difference being the replacement of the gradient by the subgradient [36], [32]. Thus the iterative scheme is again

$$x_{k+1} = x_k - \alpha_k g_k,$$

where g_k is an element of $\partial f(x_k)$, the subdifferential of f at point x_k. Based on (1.4) and (1.6), we propose the following choice of the stepsize:

$$\alpha_k = \frac{\|\delta_k\|^2}{2(f(x_{k-1}) - f(x_k) + g_k^\top \delta_k)}.$$

In classic subgradient scheme the stepsize α_k is usually put in the form $\alpha_k = \dfrac{t_k}{\|g_k\|}$ and the more popular settings of the sequence $\{t_k\}$ ensuring convergence are:

i) Constant step: $t_k = h$.

ii) Polyak stepsize:

$$t_k = \frac{f(x_k) - f^*}{\|g_k\|}, \tag{2.1}$$

where f^* is the optimal value of f or, possibly, a lower bound.

iii) Nonsummable diminishing: $t_k \to 0$ and $\displaystyle\sum_{k=1}^{\infty} t_k = \infty$.

iv) Square summable but not summable: $t_k \to 0$, $\displaystyle\sum_{k=1}^{\infty} t_k^2 < \infty$ and $\displaystyle\sum_{k=1}^{\infty} t_k = \infty$.

Our choice corresponds to the sequence

$$t_k = \frac{\|\delta_k\|^2 \|g_k\|}{2(f(x_{k-1}) - f(x_k) + g_k^\top \delta_k)}. \tag{2.2}$$

We state now the basic scheme of our Nonsmooth Barzilai-Borwein (NSBB) algorithm as follows.

NSBB algorithm

Step 1 Initialization. Select two starting points $x_0, x_1 \in \mathbb{R}^n$. Calculate $f(x_0)$. Set $f_{best} = f(x_0)$ Fix the maximum number of allowed iterations k_{max} and the stepsize safeguards $0 < t_m < t_M$. Set the iteration counter $k = 1$.

Step 2 Calculate $f(x_k)$ and $g_k \in \partial f(x_k)$. If $f(x_k) < f_{best}$ then set $f_{best} = f(x_k)$. Calculate t_k. If $t_k < t_m$ then set $t_k = t_m$. If $t_k > t_M$ then set $t_k = t_M$.

Step 3 Calculate

$$x_{k+1} = x_k - \frac{t_k}{\|g_k\|} g_k$$

Set $k = k + 1$. If success occurs in a termination test STOP, else return to Step 2.

Some discussion on NSBB algorithm is presented in what follows. We recall first that the subgradient method is nonmonotone, as it is not guaranteed that $f(x_{k+1}) < f(x_k)$. It is based, instead, on the property of any anti-subgradient direction of being a *minimum approaching* one [19]. In fact it is possible to get closer to a minimiser while moving along it, provided the stepsize is not too large. The need of introducing appropriate safeguards is thus motivated by the heuristic nature of our stepsize,which does not prevent the choice of *too large* stepsizes. They are, in fact, likely to occur whenever the denominator in formula (2.2), which is a measure of the linearization error between x_k and x_{k-1}, is small.

We give now a proof of the termination of the subgradient method when it is assumed that t_k is in a given interval $[t_m, t_M]$. The proof is an adaptation of the historical convergence proof of the subgradient method with constant stepsize provided by Shor ([36], Theorem 2.1). It is based on the following observations. Let $U_k = \{x \mid f(x) = f(x_k)\}$ be the contour line passing through x_k, and $L_k = \{x \mid g_k^\top (x - x_k) = 0\}$ be the supporting hyperplane at x_k to the level set $S_k = \{x \mid f(x) \leq f(x_k)\}$, with normal g_k. Consider now (see Figure 2.1) $a_k(x^*) = \|x^* - x_P^*\|$, the distance of any point $x^* \in M$ from its projection x_P^* onto L_k. Convexity of f implies $f(x_P^*) \geq f(x_k)$ and, from continuity, it follows $a_k(x^*) \geq b_k(x^*) = \|x^* - x_L^*\|$, where x_L^* is the intersection of U_K with the segment joining the points x^* and x_P^*. Note also that $b_k(x^*)$ is an upper bound on $dist(x^*, U_k)$, the distance of x^* from contour line U_k. On the other hand it is easy to verify that

$$a_k(x^*) = \frac{g_k^\top (x_k - x^*)}{\|g_k\|},$$

and, finally, we obtain

$$dist(x^*, L_k) \leq b_k(x^*) \leq a_k(x^*) = \frac{g_k^\top (x_k - x^*)}{\|g_k\|}. \tag{2.3}$$

We now state the following result.

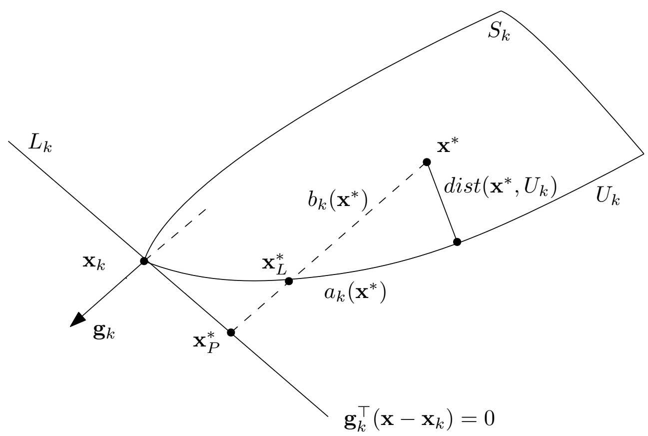


Figure 2.1: Convergence of the subgradient method

THEOREM 2.1. *Let f be a convex function and let M^*, the set of minima, be non empty. Assume that, starting from any point x_1, a sequence of points $\{x_k\}$ is generated by the following iterative scheme*

$$x_{k+1} = x_k - \frac{t_k}{\|g_k\|}, \quad k = 1, 2, \ldots \tag{2.4}$$

with $t_k \in [t_m, t_M]$, $t_m > 0$. Then, for every $\epsilon > 0$ and $x^* \in M^*$, there exist a point $\bar{x}$ and an index $\bar{k}$ such that

$$\|\bar{x} - x^*\| < \frac{t_M}{2}(1+\epsilon)$$

and

$$f(x_{\bar{k}}) = f(\bar{x}).$$

Proof. From (2.3) and (2.4) it follows that

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 = &\quad \|x_k - x^* - t_k \frac{g_k}{\|g_k\|}\|^2 \\
= &\quad \|x_k - x^*\|^2 + t_k^2 - 2t_k \frac{g_k^\top (x_k - x^*)}{\|g_k\|} \\
= &\quad \|x_k - x^*\|^2 + t_k^2 - 2t_k a_k(x^*) \\
\leq &\quad \|x_k - x^*\|^2 + t_k^2 - 2t_k b_k(x^*).
\end{aligned}$$

Now, suppose for a contradiction that $b_k(x^*) \geq t_M(1+\epsilon)/2$ for every k. By repeatedly applying the inequality (2.5), we have that for every k it holds

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 \leq \quad & \|x_1 - x^*\|^2 + \sum_{i=1}^{k} t_i^2 - 2\sum_{i=1}^{k} t_i b_i(x^*) \\
\leq \quad & \|x_1 - x^*\|^2 + \sum_{i=1}^{k} t_i^2 - t_M(1+\epsilon)\sum_{i=1}^{k} t_i \\
\leq \quad & \|x_1 - x^*\|^2 + t_M\sum_{i=1}^{k} t_i - t_M(1+\epsilon)\sum_{i=1}^{k} t_i \\
\leq \quad & \|x_1 - x^*\|^2 - \epsilon k t_m t_M
\end{aligned}$$

which contradicts $\|x_{k+1} - x^*\|^2 \geq 0$ for all k. □

REMARK 2.2. *Convergence of NSBB proved in the above theorem is rather weak. It can be strengthened if we let the interval $[t_m, t_M]$ be dynamically updated, that is if we assume $t_k \in [t_m^{(k)}, t_M^{(k)}]$. Under such hypothesis, NSBB can be seen as a particular case of the subgradient algorithm with nonlinear projections (SANP) discussed in [7], when the distance-like function adopted is based on the squared Euclidean norm. If we assume that both sequences $\{t_m^{(k)}\}$ and $\{t_M^{(k)}\}$ are nonsummable diminishing (with $t_m^{(k)} < t_M^{(k)}$, $\forall k$), the convergence properties described in [7], Theorem 4.1., hold true for NSBB as well.*

3. Applications of the method. We present in this Section the results of the implementation of our algorithm. We have considered two types of tests, a nonsmooth optimization problem coming from the application of Lagrangian relaxation [21] to a network optimization problem and a set of academic examples widely used in numerical nonsmooth optimization [4].

3.1. Lagrangian relaxation of MSTC. We focus here on a Lagrangian relaxation scheme [15], [20] applied to a graph optimization problem known in literature as the Minimum Spanning Tree Problem with Conflicting Edge Pairs (MSTC), a variant of the Minimum Spanning Tree in which there are mutually exclusive edges. More in detail, given an undirected and edge-weighted graph $G = (V, E, P)$, where $P \subseteq E \times E$ represents the set of *conflict edge pairs*, MSTC consists of finding a minimum spanning tree of G with no edges in conflict. In the following, we denote by n and m the cardinality of the vertex set V and edges set E of G, respectively. The set of conflict edge pairs P is formally defined as follows:

$P = \{\{e_i, e_j\} : e_i, e_j \in E,\ e_i \text{ and } \mathrm{e_j} \text{ cannot co} - \text{exist in the spanning tree}\}.$

To each edge $e_i \in E$ is associated a weight w_{e_i} and the set of edges $\chi(e_i)$ in conflict with it. A *spanning tree* $T(V_T, E_T)$ of G is a connected subgraph of G where $V_T = V$, $E_T \subseteq E$ and $|E_T| = n - 1$. The weight of T is equal to $W(T) = \sum_{e_i \in E_T} w_{e_i}$ and T is classified *conflict–free* if and only if there are not conflicting edges in E_T.

We now report the classical Subtour Elimination formulation of the Minimum Spanning Tree problem [10] in which a binary variable x_e is associated with each edge of G with the following meaning:

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is selected for the conflict–free tree;} \\ 0 & \text{otherwise.} \end{cases}$$

MSTC is as follows.

$$z^* = \min \sum_{e \in E} w_e x_e \tag{3.1}$$

$$\text{subject to:} \tag{3.2}$$

$$\sum_{e \in E} x_e = |V| - 1, \tag{3.3}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad S \subset V, \, |S| \geq 3, \tag{3.4}$$

$$x_{e_i} + x_{e_j} \leq 1, \quad \{e_i, e_j\} \in P, \tag{3.5}$$

$$x_e \in \{0, 1\}, \quad e \in E. \tag{3.6}$$

The objective function (3.1) minimizes the weight of the spanning tree. Constraint (3.3) assures that the solution contains exactly $n-1$ edges while constraints (3.4) are the classical subtour elimination ones. Finally, (3.5) guarantee that two edges in conflict cannot belong to the solution. Constraints (3.6), finally, are variable restrictions.

We adopt the same relaxation scheme as in [10] (a different relaxation approach, leading to a NP-hard relaxed problem, has been recently introduced in [35]). Here the conflict constraints (3.5) are relaxed via the Lagrangian multipliers $\lambda_{ij} \geq 0$, $\{e_i, e_j\} \in P$ (grouped into the vector λ of appropriate dimension). We come out with the relaxed problem $LR(\lambda)$:

$$z(\lambda) = \min \sum_{e \in E} w_e x_e + \sum_{\{e_i, e_j\} \in P} \lambda_{ij} (x_{e_i} + x_{e_j} - 1) \tag{3.7}$$

$$\text{subject to:} \tag{3.8}$$

$$\sum_{e \in E} x_e = |V| - 1, \tag{3.9}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \qquad S \subset V, |S| \geq 3, \tag{3.10}$$

$$x_e \in \{0, 1\}, e \in E. \tag{3.11}$$

Apart the constant term $(-\sum_{\{e_i,e_j\}\in P} \lambda_{ij})$, problem (3.7)–(3.11) is a classical minimum spanning tree one having the following edge weights:

$$\tilde{w}_{e_i}(\lambda) = \begin{cases} w_{e_i} & \text{if } \chi(e_i) = \emptyset \\ w_{e_i} + \sum_{e_j \in \chi(e_i)} \lambda_{ij} & \text{otherwise} \end{cases}$$

Function $z(\lambda)$, referred to as the Lagrangian function, provides a lower bound on the optimal value of MSTC.

The Lagrangian dual (LD) problem, aimed at finding the *best* lower bound, is defined as:

$$z_{LD} = \max_{\lambda \geq 0} z(\lambda). \tag{3.12}$$

The Lagrangian dual is a maximization problem where the objective function is non-smooth, in particular it is concave and piecewise affine. We have applied to it the method described in Section 2.

A subgradient $g(\lambda)$ of $z(\lambda)$ can be easily calculated once an optimal solution $x(\lambda)$ to the relaxed problem is available. The generic component of $g(\lambda)$ is:

$$g_{ij}(\lambda) = x_{e_i}(\lambda) + x_{e_j}(\lambda) - 1, \qquad (e_i, e_j) \in P, \tag{3.13}$$

thus iteration k of any subgradient method, taking into account the non negativity constraints, consists in updating the Lagrangian multipliers as follows:

$$\lambda_{ij}^{(k+1)} = \max\left(0, \lambda_{ij}^{(k)} + \alpha_k g_{ij}(\lambda^{(k)})\right), \qquad (e_i, e_j) \in P. \tag{3.14}$$

3.1.1. Computational results for MSTC. In this section, we describe the results of the NSBB algorithm on the Lagrangian dual of MSTC. The algorithm has been coded in C++ using the LEMON graph library [14]. All tests were performed on a machine (iMac mid 2011) with an Intel Core i7-2600 3.4 GHz processor and 8 GB of RAM.

Formula (2.2) for calculation of t_k has been implemented in the form

$$t_k = \frac{\|\delta_k\|^2 \|g(\lambda^{(k)})\|}{\epsilon + 2(z(\lambda^{(k)}) - g(\lambda^{(k)})^\top(\lambda^{(k)} - \lambda^{(k-1)}) - z(\lambda^{(k-1)}))}, \tag{3.15}$$

to avoid possible occurrence of zero denominator (see Remark 1.1). We have embedded into the basic algorithmic scheme the following stopping conditions at Step 3:

- $k \leq k_{max} = 500$;
- $\|\lambda^{(k)} - \lambda^{(k-1)}\| < \theta$,

Parameter	Considered values	Target
ϵ	{0.00001, 0.0001, 0.001}	0.00001
t_m	{0.000001, 0.000005, 0.00001}	0.000001
θ	{0.001, 0.01, 0.1}	0.001

Table 3.1: Parameter settings

where k is the iteration counter. The value of t_M has been dynamically updated in the form $t_M^{(k)} = \dfrac{10}{\log(k+1)}$.

All parameters of NSBB have been tuned by the IRACE package [24], an automatic configuration tool for parameter setting. Table 3.1 reports, for each parameter, the set of tested values (*Considered values*) and the corresponding target value (*Target*) in the best configuration found by IRACE.

The computational tests have been carried out on the instances proposed in [37] (*dataset 1*) and in [11] (*dataset 2*). The instances of the first dataset are classified into two types: *type 1* (may have no conflict–free solution) and *type 2* (there is at least one conflict–free solution). The instances of dataset 2 are referred to as *type 3* instances and, for all of them, the presence of a conflict-free solution is guaranteed. The datasets are available here: `http://www.dipmat2.unisa.it/people/carrabs/www/` or, alternatively, upon request to the authors. Readers can refer to [37] and to [11] to have more information concerning the generation and the characteristics of the instances of these datasets.

We compare the solutions found by NSBB on all instances with those provided by two other classic algorithms for *black box* convex functions (no special structure required). We consider the classic subgradient with nonsummable diminishing and with Polyak stepsize sequences, respectively. In order to carry out a fair comparison, we have extended the stopping conditions of NSBB to all the algorithms.

To report the results in an easy and compact form, we use the graphics of Figures 3.1, 3.2 and 3.3. These graphics were generated by using both the detailed results presented and commented in the Appendix, and the ones given in [10]. Notice that the three algorithms are compiled and executed on the same machine and then their CPU times are directly comparable.

We start our analysis by evaluating the quality of the lower bounds found by Subgradient, Polyak, and NSBB algorithms. To this end, for each instance, we computed first the best lower bound as the maximum one among the lower bounds returned by the three algorithms and from those presented in [10]. Then, we compared the lower bound of Subgradient, Polyak, and NSBB with such best lower bound. The results of this comparison are shown in the charts of Figure 3.1, which was generated by considering the results on all the instances from dataset 1 and 2.

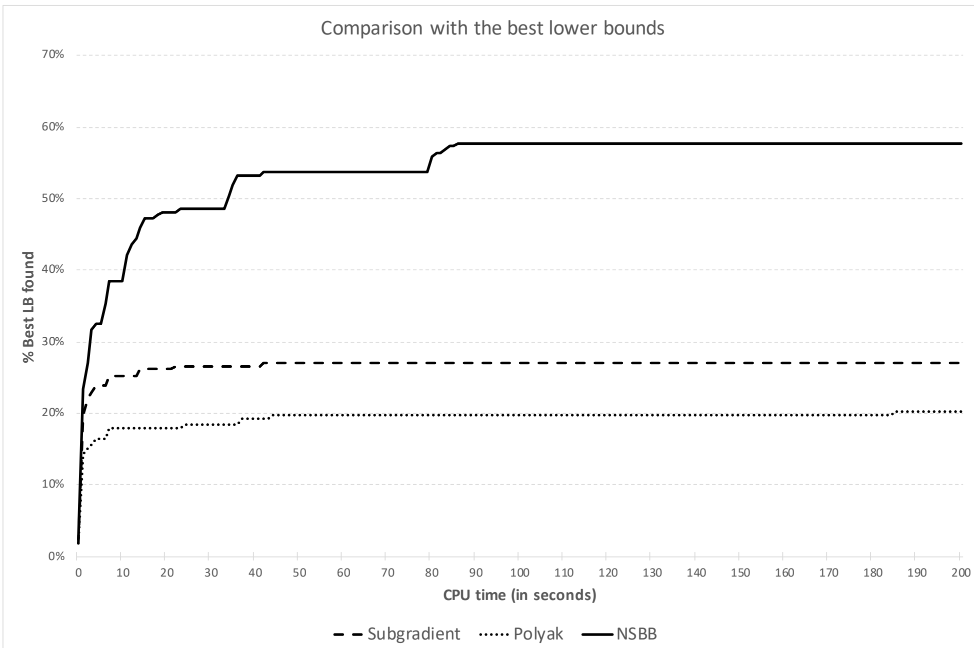


Figure 3.1: Percentage of best lower bounds found by the algorithms within the computational time reported on the x-axis

The horizontal axis in Figure 3.1 reports the computational time in seconds and the vertical one shows the percentage of best lower bounds found within that time. This means that the faster the growth of a curve, the better the performance. The chart in Figure 3.1 certifies the effectiveness and good performance of NSBB that finds the best lower bound for the ∼58% of the instances in 86 seconds. The other two algorithms are significantly less effective as Subgradient and Polyak find only the ∼27% and ∼20% of best lower bounds, respectively.

As for the time needed to reach the best result, we observe that subgradient requires about 43 seconds to reach its peak (27%), while the same result is gained by NSBB in 2 seconds only. Moreover, NSBB requires only one second to find the 20% of the best lower bounds, while Polyak reaches this peak after 185 seconds.

Figure 3.1 is about the percentage of instances where each algorithm found the best lower bound. Additional information is reported in Figure 3.2. The horizontal axis reports the percentage gap from the best lower bound at the end of the computation, whereas the vertical one shows the percentage of instances for which the

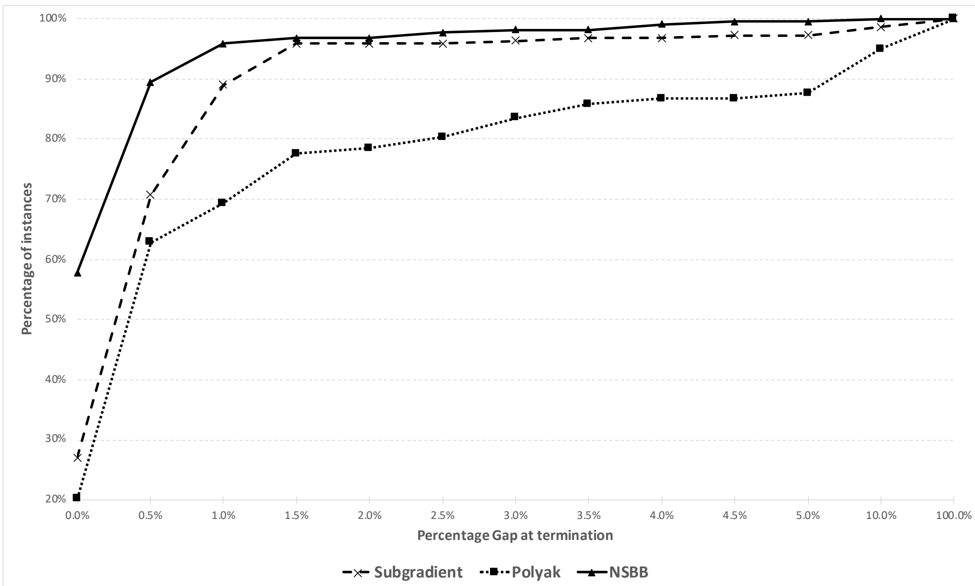


Figure 3.2: Cumulative chart of the percentage of instances solved within a given gap at termination.

percentage gap returned by the algorithm at termination is lower than or equal to the value reported on the horizontal axis. This means that the faster the growth of a curve, the better the effectiveness of the corresponding algorithm.

For x=0%, the curves of Figure 3.2 show the percentage of instances for which the algorithms found the best lower bounds. These percentages are equal to 27%, 20%, and 58% for Subgradient, Polyak, and NSBB algorithms, respectively, and they coincide with their peaks already observed in Figure 3.1.

The solid curve of NSBB shows that the percentage gap from the best lower bound is lower than or equal to 0.5%, for ∼90% of the instances, and to 1% for ∼96% of them. Less effective are the other two algorithms because Subgradient finds a lower bound with a gap within 0.5% for 71% of the instances whereas Polyak for 63% of them. These are values significantly lower than the ones obtained by NSBB.

By considering the percentage gap of 1%, the lower bounds of Subgradient are within this threshold for 89% of the instances whereas the ones of Polyak for the 69% of instances. Summarizing, the results of Figures 3.1 and 3.2 have proven that the lower bounds provided by NSBB coincide with the best lower bounds or are very close to them for most of the instances of datasets 1 and 2.

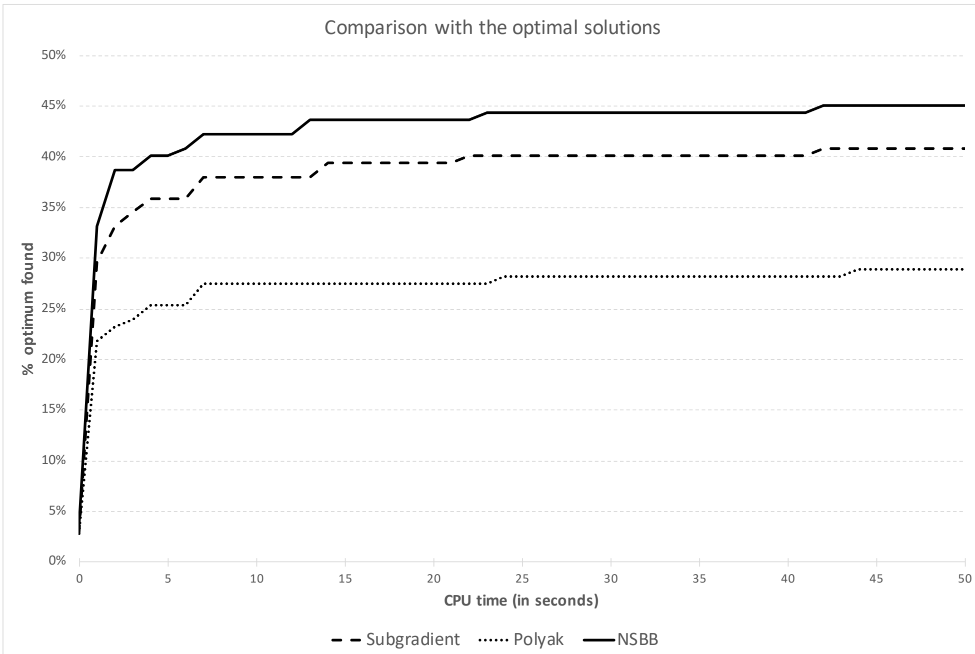


Figure 3.3: Percentage of optimal solutions found by the algorithms within the computational time reported on the x-axis

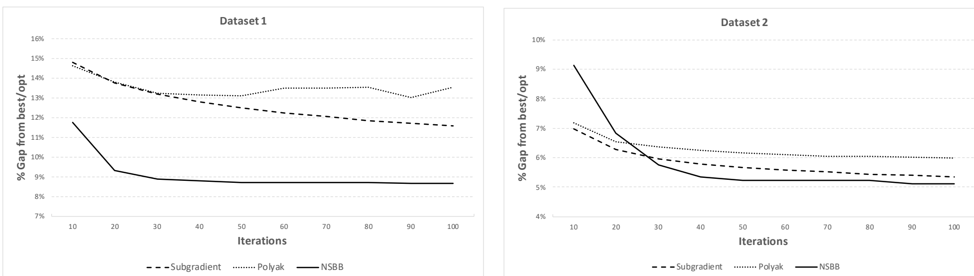
To further investigate the effectiveness of the three algorithms, we compare in Figure 3.3 the solutions they provide with the optimal ones. The chart is built by using the 144 instances, from both dataset 1 and 2, for which the optimal solution is known. Thus the y-axis is now associated with the percentage of optimal solutions found by the algorithms.

The results shown in Figure 3.3 reveal again that NSBB performs best but Subgradient obtains similar results. Indeed, the peaks of NSBB and Subgrdient are equal to ∼45% and ∼41%, respectively, and they are obtained in 42 seconds. However, NSBB reaches the peak of Subgradient (∼41%) in 6 seconds only. Finally, Polyak finds the optimal solution for 29% of instances and this result is obtained in 44 seconds.

The stepsize rule introduced in this paper is of heuristic nature and, as discussed in previous Sections, the use of safeguards is necessary. An experimental validation of the approach can be provided by assuming as an indicator the number of times (iterations) when the NSBB stepsize is *active*, that is it falls inside the safeguard interval ($t_k \in (t_m^{(k)}, t_M^{(k)})$). Such value is indicated as *Range* and is reported in Table 3.2

	ITER	Range	Range%
dataset [37]	47.79	44.95	94.05%
dataset [11] (small)	96.07	87.96	91.56%
dataset [11] (large)	77.90	65.09	83.55%

Table 3.2: New stepsize occurrences.



(a) (b)

Figure 3.4: Behaviour of the three algorithms, according to the maximum number of iterations, on (a) the first dataset and (b) the second dataset.

for the various groups of instances, together with the average number of iterations *ITER*.

The average percentage (Column "Range%") appears satisfactorily high and indicates that relatively seldom the safeguards enter into play. Note also that column ITER reveals that the average number of iterations is significantly smaller than the maximumm allowed (500).

Finally, to better highlight the performance of the three algorithms in terms of number of iterations, we plot in Figure 3.4 the percentage gap from the best-known solution value for different values of the maximum number of iterations, in the range $[10, 100]$, with step of 10.

Note that faster decrease indicates better effectiveness. We observe that for the first data set NSBB is uniformly the most effective, while for the second dataset it becomes the best as the number of considered iteration is sufficiently large. A more detailed discussion of the results is given in the Appendix.

3.2. Academic examples. We have considered the following seven test functions for unconstrained convex nonsmooth optimization [27],[4]. For each function we report the standard starting point here indicated as $x^{(0)}$, the minimum x^* and the optimal value f^*.

1. *Dem–Mal*: $f(x) = \max\{5x_1 + x_2, -5x_1 + x_2, x_1^2 + x_2^2 + 4x_2\}$; $x^{(0)} = (1, 1)$; $x^* = (0, -3)$; $f^* = -3$.
2. *Mifflin*: $f(x) = -x_1 + 20 \max\{x_1^2 + x_2^2 - 1, 0\}$; $x^{(0)} = (0.8, 0.6)$; $x^* = (1, 0)$; $f^* = -1$.
3. *LQ*: $f(x) = \max\{-x_1 - x_2, -x_1 - x_2 + x_1^2 + x_2^2 - 1\}$; $x^{(0)} = (-0.5, -0.5)$; $x^* = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$; $f^* = -\sqrt{2}$.
4. *MAXQ*: $f(x) = \max\limits_{1 \leq i \leq 20} \{x_i^2\}$; $x_i^{(0)} = 0$, $i = 1, \dots, 10$; $x_i^{(0)} = -i$, $i = 11, \dots, 20$; $x^* = (0, \dots, 0)$; $f^* = 0$.
5. *QL*: $f(x) = \max\limits_{1 \leq i \leq 3} f_i(x)$; $f_1(x) = x_1^2 + x_2^2$; $f_2(x) = x_1^2 + x_2^2 + 10(-4x_1 - x_2 + 4)$, $f_2(x) = x_1^2 + x_2^2 + 10(-x_1 - 2x_2 + 6)\}$; $x^{(0)} = (-1, 5)$; $x^* = (1.2, 2.4)$; $f^* = 7.2$.
6. *CB2*: $f(x) = \max\{x_1^2 + x_2^4, (2 - x_1)^2 + (2 - x_2)^2, 2e^{(-x_1 + x_2)}\}$; $x^{(0)} = (1, -0.1)$; $x^* = (1.1392286, 0.899365)$; $f^* = 1.9522245$.
7. *CB3*: $f(x) = \max\{x_1^4 + x_2^2, (2 - x_1)^2 + (2 - x_2)^2, 2e^{(-x_1 + x_2)}\}$; $x^{(0)} = (2, 2)$; $x^* = (1, 1)$; $f^* = 2$.

In dealing with the academic examples we have set the following stopping conditions at Step 3:

- $k \leq k_{max} = 1000$;
- $\|x_k - x^*\| < \eta$ OR $f(x_k) - f^* < \eta$, with $\eta = 0.01$.

The interval (t_m, t_M) has been dynamically updated in the form $(\frac{0.0001}{k}, \frac{1}{k})$ (see Remark 2.2). As for calculation of t_k at Step 2, taking into account Remark 1.1, we have set

$$t_k = \begin{cases} t_{k-1}, & \text{if } (f(x_{k-1}) - f(x_k) + g_k^\top \delta_k) \leq \eta = 0.001, \\ \dfrac{\|\delta_k\|^2 \|g_k\|}{2(f(x_{k-1}) - f(x_k) + g_k^\top \delta_k)}, & \text{otherwise.} \end{cases}$$

We have compared NSBB with four versions of classic subgradient method. The first one, referred to as Polyak, is based on stepsize (2.1). The remaining three implement diminishing nonsummable sequences, by setting $t_k = 1/k$, $t_k = 1/\sqrt{k}$, $t_k = 1/\log(k + 1)$; they are referred to as harmonic, square root and logarithmic sequences, respectively. The termination tests are the same as for NSBB.

The results in terms of number of function–subgradient evaluations are in table 3.3. As for MSCT, the column "Range" reports the number of times the calculated stepsize has been within the prefixed range $(t_m^{(k)}, t_M^{(k)})$. The "*" symbol indicates that t_k has been allowed to range in the interval $(0, \infty)$. The total percentage of $t_k \in (t_m^{(k)}, t_M^{(k)})$ is 74%.

Our stepsize rule appears rather effective and reliable compared with the classic ones, which provide results somehow erratic, as several times the max number of iteration is reached with no satisfaction of the stopping criterion.

Function	# variables	NSBB	Range	Polyak	Subg-harm.	Subg-square	Subg-log
Dem–Mal	2	12	9	76	> 1000	82	62
Mifflin	2	28	24	103	18	15	> 1000
LQ	2	3*	–	2	18	3	10
Maxq	20	46*	–	139	> 1000	> 1000	> 1000
QL	2	27	20	> 1000	> 1000	93	> 1000
CB2	2	34	20	132	14	14	> 1000
CB3	2	22	19	11	20	83	947

Table 3.3: Academic test problems. Function–subgradient evaluations

4. Conclusions. We have introduced a two–point–based stepsize rule in subgradient method for convex nonsmooth optimization. Our rule coincides with Barzilai–Borwein method when applied to a convex quadratic. The numerical experience both on a Lagrangian relaxation example and on some academic test problems has revealed satisfactory.

5. Declaration of interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6. Acknowledgement. The research of F. Carrabs was partially supported by the project "SEcurity and RIghts in the CyberSpace" SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. This support is gratefully acknowledged.

7. Appendix: Detailed computational results. We report here the detailed results of the comparison among the Subgradient, Polyak and NSBB algorithms. The instances of dataset 2 are partitioned into two groups: the *small* instances, having at most 50 nodes, and the *big* ones, having more than 50 nodes. In implementing Polyak's stepsize we have taken as f^* the value of the feasible solution provided by the heuristic used in [10].

In Table 7.1 we compare the solutions found by NSBB with the ones found by the classic subgradient with nonsummable diminishing and with Polyak stepsize sequences, respectively, on the instances of dataset 1 [37]. Under the instance heading, we report the following characteristics of the instances: a numerical identifier (id), the number of nodes (n), edges (m) and conflict pairs (p). Column Opt reports the optimal solution value or the best known solution value whenever the "*" symbol is present. Then are reported the lower bound (LB), the computational time ($Time$) (in seconds) and the percentage gap (Gap) from the Opt column value for Subgradient, Polyak and NSBB algorithms. The Gap value is computed by using the formula: $100 \times \frac{Opt-LB}{Opt}$. At the bottom, AVG shows the average of computation time and of percentage gap for each algorithm. The last row indicates that NSBB is the fastest

	Instance					Subgradient			Polyak			NSBB		
	ID	n	m	p	Opt	LB	Time	Gap	LB	Time	Gap	LB	Time	Gap
Type 1 Feasible	1	50	200	199	708	703.58	0.23	0.62%	704.88	0.31	0.44%	705.50	0.12	0.35%
	2	50	200	398	770	755.78	0.29	1.85%	759.15	0.41	1.41%	759.34	0.13	1.38%
	3	50	200	597	917	837.56	0.32	8.66%	819.28	0.48	10.66%	859.65	0.20	6.25%
	4	50	200	995	1324	936.70	0.40	29.25%	850.26	0.73	35.78%	940.63	0.18	28.96%
	5	100	300	448	4041	3864.55	0.77	4.37%	4029.57	1.04	0.28%	4007.11	0.51	0.84%
	6	100	300	897	5658	4543.19	1.28	19.70%	3132.00	2.03	44.64%	4847.60	0.88	14.32%
	7	100	500	1247	4275	4018.28	2.04	6.01%	4266.63	2.85	0.20%	4212.57	1.72	1.46%
	8	100	500	2495	5997	4705.01	2.30	21.54%	4896.23	4.84	18.36%	5029.08	1.80	16.14%
	9	100	500	3741	7665*	5053.21	2.52	34.07%	3742.73	5.40	51.17%	5218.64	1.62	31.92%
Type 1 F.Unknown	10	200	600	1797	15029*	10776.40	4.54	28.30%	7386.00	10.31	50.86%	11921.60	3.79	20.68%
	11	200	800	3196	22110*	16352.50	7.98	26.04%	11939.00	19.30	46.00%	18367.30	6.75	16.93%
Type 2	12	50	200	3903	1636	1038.44	1.24	36.53%	931.17	2.12	43.08%	1020.74	0.52	37.61%
	13	50	200	4877	2043	1106.81	1.61	45.82%	905.49	2.42	55.68%	1099.63	0.84	46.18%
	14	50	200	5864	2338	2331.73	0.75	0.27%	2338.00	0.69	0.00%	2338.00	0.66	0.00%
	15	100	300	8609	7434	7434.00	0.24	0.00%	7434.00	0.25	0.00%	7434.00	0.24	0.00%
	16	100	300	10686	7968	7968.00	0.17	0.00%	7968.00	0.18	0.00%	7968.00	0.17	0.00%
	17	100	300	12761	8166	8166.00	0.11	0.00%	8166.00	0.11	0.00%	8166.00	0.11	0.00%
	18	100	500	24740	12652	5528.19	9.66	56.31%	4743.83	19.36	62.51%	5553.21	4.78	56.11%
	19	100	500	30886	11232	5655.70	16.60	49.65%	5040.59	24.55	55.12%	5679.48	11.12	49.43%
	20	100	500	36827	11481	11481.00	21.68	0.00%	11481.00	23.15	0.00%	11481.00	22.23	0.00%
	21	200	400	13660	17728	17728.00	0.08	0.00%	17728.00	0.04	0.00%	17728.00	0.04	0.00%
	22	200	400	17089	18617	18617.00	0.20	0.00%	18617.00	0.05	0.00%	18617.00	0.05	0.00%
	23	200	400	20470	19140	19140.00	0.05	0.00%	19140.00	0.05	0.00%	19140.00	0.05	0.00%
	24	200	600	34504	20716	20716.00	1.48	0.00%	20716.00	1.57	0.00%	20716.00	1.54	0.00%
	25	200	600	42860	18025	18025.00	0.74	0.00%	18025.00	0.77	0.00%	18025.00	0.75	0.00%
	26	200	600	50984	20864	20864.00	0.57	0.00%	20864.00	0.58	0.00%	20864.00	0.58	0.00%
	27	200	800	62625	39895	39895.00	41.16	0.00%	39895.00	43.40	0.00%	39895.00	41.78	0.00%
	28	200	800	78387	37671	37671.00	6.36	0.00%	37671.00	6.66	0.00%	37671.00	6.22	0.00%
	29	200	800	93978	38798	38798.00	3.12	0.00%	38798.00	3.29	0.00%	38798.00	3.06	0.00%
	30	300	600	31000	43721	43721.00	0.10	0.00%	43721.00	0.10	0.00%	43721.00	0.09	0.00%
	31	300	600	38216	44267	44261.20	2.54	0.01%	44267.00	0.13	0.00%	44267.00	0.15	0.00%
	32	300	600	45310	43071	43071.00	0.12	0.00%	43071.00	0.13	0.00%	43071.00	0.12	0.00%
	33	300	800	59600	43125	43125.00	0.17	0.00%	43125.00	0.17	0.00%	43125.00	0.16	0.00%
	34	300	800	74500	42292	42292.00	0.20	0.00%	42292.00	0.20	0.00%	42292.00	0.19	0.00%
	35	300	800	89300	44114	44114.00	0.22	0.00%	44114.00	0.22	0.00%	44114.00	0.22	0.00%
	36	300	1000	96590	71562	71562.00	6.29	0.00%	71562.00	6.48	0.00%	71562.00	6.30	0.00%
	37	300	1000	120500	76345	76345.00	3.71	0.00%	76345.00	3.94	0.00%	76345.00	3.66	0.00%
	38	300	1000	144090	78880	78880.00	1.89	0.00%	78880.00	2.09	0.00%	78880.00	1.91	0.00%
AVG							**3.78**	**9.71%**		**5.01**	**12.53%**		**3.30**	**8.65%**

Table 7.1: Comparison on the instances of the first dataset.

and most effective algorithm with an average time equal to 3.30 seconds and a gap from the best known solution equal to 8.65%. Subgradient is slightly slower than NSBB and less effective with an average gap equal to 9.71%. The highest average time and gap are those obtained by Polyak with a computational time of 5.01 seconds and an average gap equal to 12.53%.

Table 7.2 shows the computational results of the three algorithms on the small instances of dataset 2 [11]. The AVG row shows that the three algorithms are very fast and effective on such instances. Indeed, the computational time is lower than 1 second for both NSBB and Subgradient and it is equal to 1.14 seconds for Polyak. NSBB, however, is again the fastest algorithm. As for effectiveness, the Gap values of Subgradient and NSBB are very close (3.55% and 3.57%, respectively). For Polyak the Gap value is slightly bigger (4.52%). Gap column indicates that the number of conflict pairs p is the parameter that mainly affects the effectiveness of the three algorithms. Indeed, for fixed number of nodes and edges, the Gap values of the three

	Instance						Subgradient			Polyak			NSBB		
	ID	n	m	p	s	Opt	LB	Time	Gap	LB	Time	Gap	LB	Time	Gap
	51	25	60	18	1	347	347.00	0.03	0.00%	347.00	0.00	0.00%	347.00	0.00	0.00%
	52	25	60	18	7	389	389.00	0.00	0.00%	389.00	0.00	0.00%	389.00	0.00	0.00%
	53	25	60	18	13	353	353.00	0.00	0.00%	353.00	0.00	0.00%	353.00	0.00	0.00%
	54	25	60	18	19	346	346.00	0.00	0.00%	346.00	0.00	0.00%	346.00	0.00	0.00%
	55	25	60	18	25	336	336.00	0.00	0.00%	336.00	0.00	0.00%	336.00	0.00	0.00%
	56	25	60	71	31	381	379.63	0.05	0.36%	379.59	0.06	0.37%	379.64	0.02	0.36%
	57	25	60	71	37	390	380.76	0.05	2.37%	379.83	0.07	2.61%	381.50	0.02	2.18%
	58	25	60	71	43	372	372.00	0.01	0.00%	371.99	0.04	0.00%	372.00	0.01	0.00%
	59	25	60	71	49	357	356.99	0.05	0.00%	356.76	0.06	0.07%	357.00	0.02	0.00%
	60	25	60	71	55	406	406.00	0.01	0.00%	405.99	0.02	0.00%	406.00	0.01	0.00%
	61	25	60	124	61	385	384.69	0.06	0.08%	384.78	0.07	0.06%	385.00	0.02	0.00%
	62	25	60	124	67	432	432.00	0.03	0.00%	431.95	0.07	0.01%	432.00	0.02	0.00%
	63	25	60	124	73	458	423.61	0.06	7.51%	335.00	0.10	26.86%	421.65	0.01	7.94%
	64	25	60	124	79	400	395.42	0.06	1.14%	397.01	0.09	0.75%	397.53	0.06	0.62%
	65	25	60	124	85	420	405.90	0.07	3.36%	398.46	0.11	5.13%	403.30	0.02	3.98%
	66	25	90	41	91	311	311.00	0.01	0.00%	311.00	0.01	0.00%	311.00	0.01	0.00%
	67	25	90	41	97	306	306.00	0.01	0.00%	306.00	0.01	0.00%	306.00	0.01	0.00%
	68	25	90	41	103	299	299.00	0.01	0.00%	299.00	0.01	0.00%	299.00	0.01	0.00%
	69	25	90	41	109	297	297.00	0.01	0.00%	297.00	0.01	0.00%	297.00	0.01	0.00%
	70	25	90	41	115	318	318.00	0.01	0.00%	318.00	0.01	0.00%	318.00	0.01	0.00%
	71	25	90	161	121	305	305.00	0.02	0.00%	304.99	0.03	0.00%	305.00	0.02	0.00%
	72	25	90	161	127	339	339.00	0.08	0.00%	338.90	0.11	0.03%	339.00	0.05	0.00%
	73	25	90	161	133	344	344.00	0.03	0.00%	343.74	0.09	0.07%	344.00	0.02	0.00%
	74	25	90	161	139	329	327.78	0.07	0.37%	327.83	0.11	0.36%	327.90	0.03	0.34%
	75	25	90	161	145	326	324.98	0.07	0.31%	324.95	0.11	0.32%	324.93	0.02	0.33%
	76	25	90	281	151	349	347.15	0.09	0.53%	346.76	0.12	0.64%	348.00	0.09	0.29%
	77	25	90	281	157	385	370.24	0.10	3.83%	369.01	0.15	4.15%	369.40	0.03	4.05%
	78	25	90	281	163	335	330.64	0.10	1.30%	330.53	0.14	1.33%	330.55	0.09	1.33%
	79	25	90	281	169	348	334.24	0.10	3.95%	331.69	0.16	4.69%	334.08	0.03	4.00%
	80	25	90	281	175	357	349.94	0.09	1.98%	349.24	0.15	2.17%	348.11	0.03	2.49%
	81	25	120	72	181	282	282.00	0.02	0.00%	282.00	0.02	0.00%	282.00	0.02	0.00%
	82	25	120	72	187	294	294.00	0.02	0.00%	294.00	0.02	0.00%	294.00	0.02	0.00%
	83	25	120	72	193	284	283.96	0.06	0.01%	283.99	0.04	0.00%	284.00	0.02	0.00%
	84	25	120	72	199	281	280.98	0.06	0.01%	280.98	0.04	0.01%	281.00	0.02	0.00%
	85	25	120	72	205	292	292.00	0.02	0.00%	292.00	0.02	0.00%	292.00	0.02	0.00%
	86	25	120	286	211	321	320.11	0.11	0.28%	319.66	0.15	0.42%	320.17	0.06	0.26%
	87	25	120	286	217	317	316.93	0.11	0.02%	315.99	0.15	0.32%	317.00	0.04	0.00%
	88	25	120	286	223	284	284.00	0.03	0.00%	283.81	0.14	0.07%	284.00	0.03	0.00%
	89	25	120	286	229	311	311.00	0.05	0.00%	310.61	0.15	0.12%	311.00	0.03	0.00%
	90	25	120	286	235	290	290.00	0.03	0.00%	290.00	0.03	0.00%	290.00	0.03	0.00%
	91	25	120	500	241	329	318.78	0.15	3.11%	318.30	0.22	3.25%	318.30	0.04	3.25%
	92	25	120	500	247	339	325.29	0.15	4.04%	324.45	0.23	4.29%	324.22	0.04	4.36%
	93	25	120	500	253	368	353.70	0.15	3.89%	349.55	0.28	5.01%	352.73	0.05	4.15%
	94	25	120	500	259	311	306.06	0.14	1.59%	305.80	0.20	1.67%	303.69	0.15	2.35%
Type 3	95	25	120	500	265	321	318.00	0.14	0.93%	317.64	0.22	1.05%	316.26	0.04	1.48%
Small	96	50	245	299	271	619	619.00	0.19	0.00%	618.97	0.24	0.01%	619.00	0.17	0.00%
	97	50	245	299	277	604	604.00	0.17	0.00%	603.92	0.19	0.01%	604.00	0.17	0.00%
	98	50	245	299	283	634	634.00	0.16	0.00%	634.00	0.18	0.00%	634.00	0.16	0.00%
	99	50	245	299	289	616	615.50	0.31	0.08%	615.26	0.37	0.12%	615.50	0.19	0.08%
	100	50	245	299	295	595	595.00	0.30	0.00%	594.97	0.19	0.00%	595.00	0.16	0.00%
	101	50	245	1196	301	678	667.71	0.52	1.52%	665.69	0.92	1.82%	667.84	0.51	1.50%
	102	50	245	1196	307	681	654.24	0.52	3.93%	647.40	0.96	4.93%	650.84	0.25	4.43%
	103	50	245	1196	313	709	677.11	0.53	4.50%	670.03	0.95	5.50%	676.88	0.29	4.53%
	104	50	245	1196	319	639	633.32	0.51	0.89%	633.30	0.79	0.89%	631.34	0.51	1.20%
	105	50	245	1196	325	681	657.81	0.52	3.40%	652.92	0.96	4.12%	655.37	0.27	3.76%
	106	50	245	2093	331	833*	657.46	0.69	21.07%	593.69	1.47	28.73%	659.91	0.25	20.78%
	107	50	245	2093	337	835	702.27	0.70	15.90%	640.91	1.43	23.24%	702.14	0.27	15.91%
	108	50	245	2093	343	840*	662.87	0.68	21.09%	607.75	1.29	27.65%	664.33	0.26	20.91%
	109	50	245	2093	349	836*	677.39	0.70	18.97%	627.59	1.33	24.93%	677.49	0.25	18.96%
	110	50	245	2093	355	769	691.78	0.69	10.04%	631.37	1.34	17.90%	689.94	0.29	10.28%
	111	50	367	672	361	570	570.00	0.53	0.00%	570.00	0.55	0.00%	570.00	0.53	0.00%
	112	50	367	672	367	561	561.00	0.77	0.00%	561.00	0.70	0.00%	561.00	0.55	0.00%
	113	50	367	672	373	573	573.00	0.76	0.00%	572.40	0.94	0.10%	573.00	0.55	0.00%
	114	50	367	672	379	560	560.00	0.53	0.00%	560.00	0.55	0.00%	560.00	0.52	0.00%
	115	50	367	672	385	549	549.00	0.76	0.00%	548.99	0.89	0.00%	549.00	0.54	0.00%
	116	50	367	2687	391	612	595.08	1.17	2.77%	593.58	2.16	3.01%	595.44	1.16	2.71%
	117	50	367	2687	397	615	595.33	1.16	3.20%	594.13	2.11	3.39%	594.87	0.81	3.27%
	118	50	367	2687	403	587	575.83	1.22	1.90%	575.27	2.06	2.00%	575.33	0.73	1.99%
	119	50	367	2687	409	634	606.27	1.16	4.37%	604.51	2.23	4.65%	605.91	0.70	4.43%
	120	50	367	2687	415	643	634.98	1.17	1.25%	634.42	2.13	1.33%	633.93	0.71	1.41%
	121	50	367	4702	421	726*	620.53	1.54	14.53%	590.46	3.07	18.67%	622.92	0.70	14.20%
	122	50	367	4702	427	770*	632.19	1.52	17.90%	600.12	2.83	22.06%	632.31	0.68	17.88%
	123	50	367	4702	433	786*	643.60	1.54	18.12%	611.89	3.06	22.15%	643.90	0.68	18.08%
	124	50	367	4702	439	711*	593.94	1.54	16.46%	566.44	3.25	20.33%	595.84	0.68	16.20%
	125	50	367	4702	445	764*	662.10	1.53	13.34%	638.85	3.00	16.38%	661.78	0.70	13.38%
	126	50	490	1199	451	548	548.00	1.51	0.00%	548.00	1.67	0.00%	548.00	1.22	0.00%
	127	50	490	1199	457	530	529.95	1.51	0.01%	529.16	1.68	0.16%	530.00	1.20	0.00%
	128	50	490	1199	463	549	549.00	1.49	0.00%	548.50	1.29	0.09%	549.00	1.21	0.00%
	129	50	490	1199	469	540	539.97	1.53	0.01%	540.00	1.77	0.00%	539.79	1.24	0.04%
	130	50	490	1199	475	540	540.00	1.20	0.00%	540.00	1.24	0.00%	540.00	1.20	0.00%
	131	50	490	4793	481	594	583.44	2.20	1.78%	583.30	3.90	1.80%	583.83	1.80	1.71%
	132	50	490	4793	487	579	558.24	2.20	3.59%	558.21	4.08	3.59%	558.48	1.36	3.54%
	133	50	490	4793	493	589	576.67	2.20	2.09%	576.87	4.05	2.06%	576.12	1.33	2.19%
	134	50	490	4793	499	577	563.78	2.20	2.29%	563.74	3.99	2.30%	564.23	1.37	2.21%
	135	50	490	4793	505	592	576.00	2.20	2.70%	575.73	3.96	2.75%	574.49	1.36	2.96%
	136	50	490	8387	511	678*	571.30	2.85	15.74%	569.43	5.50	16.01%	572.86	1.52	15.51%
	137	50	490	8387	517	651*	565.23	2.84	13.17%	564.33	5.49	13.31%	566.60	1.52	12.96%
	138	50	490	8387	523	689*	589.00	2.87	14.51%	586.66	5.74	14.85%	589.99	1.47	14.37%
	139	50	490	8387	529	682*	588.26	2.87	13.74%	569.87	5.69	16.44%	590.89	1.49	13.36%
	140	50	490	8387	535	674*	583.41	2.88	13.44%	581.35	6.60	13.75%	585.61	1.48	13.11%
AVG								**0.66**	**3.55%**		**1.14**	**4.52%**		**0.40**	**3.57%**

Table 7.2: Comparison of three algorithms on the small instances of the second dataset.

algorithms increases as p increases. As an example, if we consider the largest instances (126-140) of the table, we observe that for the instances with id from 126 to 135 the Gap value of NSBB is lower than 3%, while for the remaining (136-140), it increases from 12.96% to 15.51%.

Table 7.3 shows the computational results of the three algorithms on the large instances of the second benchmark dataset [11]. The AVG values of the last row show that in these instances NSBB is twice faster than Polyak and 25% faster than Subgradient. Moreover, NSBB is the most effective too with an average Gap value equal to 7.01%, against the 7.17% of Subgradient and 7.60% of Polyak. It is worth noting that in all the instances where a feasible solution is not known (the rows with the symbol "-'' under the Opt heading) the best lower bound is always found by NSBB.

REFERENCES

[1] Anstreicher, K.M., L.A. Wolsey, Two "well known" properties of subgradient optimization, *Mathematical Programming Series B* 120,1: 213–220, 2009.

[2] Astorino, A., M. Gaudioso and G. Miglionico, A Lagrangean relaxation approach to lifetime maximization of directional sensor networks. *Networks* 78,1: 5–16, 2021.

[3] Astorino, A., A. Frangioni, M. Gaudioso and E. Gorgone, Piecewise quadratic approximations in convex numerical optimization. *SIAM Journal on Optimization* 21: 1418–1438, 2011.

[4] Bagirov, A., N. Karmitsa, M. Mäkelä, Introduction to Nonsmooth Optimization, Springer, 2014.

[5] Bagirov, A., M. Gaudioso, N. Karmitsa, M. Mäkelä, S.Taheri, Numerical Nonsmooth Optimization. State of the Art Algorithms, Springer, 2020.

[6] Barzilai, J., J.M. Borwein, Two-point step size gradient methods, *IMA Journal of Numerical Analysis*, 8: 141–148, 1988.

[7] Beck, A., M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization, *Operations Research Letters*, 31 167–175, 2003.

[8] Bertsekas, D.P., Incremental proximal methods for large scale convex optimization, *Mathematical Programming*, 129, 2: 163–195, 2011.

[9] Carrabs, F., Cerrone, C., Pentangelo, R., A multiethnic genetic approach for the minimum conflict weighted spanning tree problem, *Networks*, 74, 2: 134–147, 2019.

[10] Carrabs, F., M. Gaudioso, A Lagrangian approach for the minimum spanning tree problem with conflicting edge pairs, *Networks*, 78, 1: 32–45, 2021.

[11] Carrabs, F., R. Cerulli, R. Pentangelo, A. Raiconi, Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach, *Annals of Operations Research*, 298, 1: 65–78, 2021.

[12] Di Puglia Pugliese, L., M. Gaudioso, F. Guerriero, and G. Miglionico, A Lagrangean-based decomposition approach for the link constrained Steiner tree problem, *Optimization Methods and Software*, 33,3: 650–670, 2018.

[13] Duchi, J., E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research* 12: 2121–2159, 2011.

[14] Dezső, B., A. Jüttner, P. Kovács, LEMON – an Open Source C++ Graph Template Library. Electronic Notes in Theoretical Computer Science, 264:23–45, 2011. `http://lemon.cs.elte.hu/trac/lemon`.

[15] Frangioni, A., About Lagrangian methods in integer optimization, *Annals of Operations Research*, 139, 1: 163–193, 2005.

[16] Frangioni, A., B. Gendron and E. Gorgone, On the computational efficiency of subgradients methods: a case study with Lagrangian bounds, *Mathematical Programming Computation* 9:

	Instance						Subgradient			Polyak			NSBB		
	ID	n	m	p	s	Opt	LB	Time	Gap	LB	Time	Gap	LB	Time	Gap
	141	75	555	1538	541	868	868.00	2.28	0.00%	868.00	2.74	0.00%	868.00	1.91	0.00%
	142	75	555	1538	547	871	871.00	2.29	0.00%	870.57	3.06	0.05%	870.51	1.84	0.06%
	143	75	555	1538	553	838	837.86	2.28	0.02%	837.54	3.16	0.05%	837.95	1.87	0.01%
	144	75	555	1538	559	855	855.00	2.28	0.00%	854.83	2.50	0.02%	855.00	1.95	0.00%
	145	75	555	1538	565	857	856.99	2.26	0.00%	856.67	2.78	0.04%	857.00	1.87	0.00%
	146	75	555	6150	571	1047*	936.37	3.26	10.57%	926.86	8.26	11.47%	939.26	2.04	10.29%
	147	75	555	6150	577	1069*	932.85	3.23	12.74%	926.85	7.25	13.30%	933.73	2.05	12.65%
	148	75	555	6150	583	1040*	906.19	3.23	12.87%	859.31	7.33	17.37%	908.25	2.11	12.67%
	149	75	555	6150	589	998*	908.95	3.24	8.92%	861.96	7.17	13.63%	909.31	2.14	8.89%
	150	75	555	6150	595	994*	892.37	3.24	10.22%	847.99	8.04	14.69%	894.89	2.16	9.97%
	151	75	555	10762	601	-	914.00	4.12		872.33	8.74		920.23	2.25	
	152	75	555	10762	607	-	939.01	4.11		897.91	8.42		944.59	2.13	
	153	75	555	10762	613	-	892.81	4.16		846.75	9.19		898.63	2.14	
	154	75	555	10762	619	-	879.64	4.18		835.43	9.04		884.89	2.04	
	155	75	555	10762	625	-	906.41	4.16		866.24	8.94		910.33	2.11	
	156	75	832	3457	631	798	797.98	6.31	0.00%	796.39	7.26	0.20%	797.62	5.53	0.05%
	157	75	832	3457	637	821	819.76	6.29	0.15%	819.78	8.29	0.15%	819.49	5.50	0.18%
	158	75	832	3457	643	816	815.14	6.32	0.10%	814.93	7.87	0.13%	815.27	5.51	0.09%
	159	75	832	3457	649	820	820.00	6.29	0.00%	820.00	6.95	0.00%	820.00	5.55	0.00%
	160	75	832	3457	655	815	814.98	6.28	0.00%	814.98	8.08	0.00%	815.00	5.45	0.00%
	161	75	832	13828	661	903*	826.23	8.34	8.50%	825.89	19.33	8.54%	828.95	5.95	8.20%
	162	75	832	13828	667	953*	854.90	8.34	10.29%	856.61	18.51	10.11%	857.14	5.85	10.06%
	163	75	832	13828	673	892*	824.54	8.35	7.56%	824.57	18.45	7.56%	826.41	5.85	7.35%
	164	75	832	13828	679	915*	837.04	8.36	8.52%	835.88	19.98	8.65%	839.47	5.96	8.25%
	165	75	832	13828	685	896*	842.82	8.33	5.94%	843.05	17.77	5.91%	844.82	5.96	5.71%
	166	75	832	24199	691	-	861.91	11.42		848.07	21.15		868.50	6.69	
	167	75	832	24199	697	-	829.30	11.41		815.45	22.22		836.01	6.36	
	168	75	832	24199	703	-	833.24	11.33		817.75	23.35		840.29	6.37	
	169	75	832	24199	709	-	854.00	11.40		839.83	23.24		860.74	6.82	
	170	75	832	24199	715	-	870.24	11.36		852.93	24.05		877.26	6.35	
	171	75	1110	6155	721	787	786.94	13.64	0.01%	786.67	17.77	0.04%	786.72	12.23	0.04%
	172	75	1110	6155	727	785	785.00	13.62	0.00%	784.16	16.21	0.11%	785.00	12.27	0.00%
	173	75	1110	6155	733	783	782.99	13.59	0.00%	782.96	18.01	0.01%	782.99	12.25	0.00%
	174	75	1110	6155	739	784	784.00	13.58	0.00%	783.74	17.22	0.03%	784.00	12.16	0.00%
	175	75	1110	6155	745	797	796.48	13.84	0.07%	795.79	17.01	0.15%	796.24	12.30	0.09%
	176	75	1110	24620	751	867*	811.77	18.34	6.37%	814.75	36.58	6.03%	813.62	13.66	6.16%
	177	75	1110	24620	757	851*	792.33	18.45	6.89%	795.16	38.79	6.56%	796.23	13.55	6.44%
	178	75	1110	24620	763	892*	803.15	18.45	9.96%	805.92	37.92	9.65%	806.20	13.68	9.62%
	179	75	1110	24620	769	864*	802.46	20.01	7.12%	805.35	36.97	6.79%	805.33	13.53	6.79%
	180	75	1110	24620	775	882*	799.34	18.75	9.37%	801.15	37.76	9.17%	801.48	13.59	9.13%
	181	75	1110	43085	781	-	809.48	26.26		803.49	49.26		815.13	14.76	
	182	75	1110	43085	787	-	794.70	26.36		790.44	51.58		802.39	14.77	
	183	75	1110	43085	793	1194*	818.63	26.90	31.44%	814.84	46.77	31.76%	824.20	17.36	30.97%
	184	75	1110	43085	799	-	790.76	27.22		785.92	50.83		797.46	14.80	
Type 3	185	75	1110	43085	805	-	797.44	26.76		792.80	51.48		804.37	18.43	
Large	186	100	990	4896	811	1119	1117.96	10.68	0.09%	1117.57	15.76	0.13%	1117.63	9.52	0.12%
	187	100	990	4896	817	1137	1134.52	10.68	0.22%	1134.41	14.56	0.23%	1134.06	9.39	0.26%
	188	100	990	4896	823	1113	1112.20	10.70	0.07%	1112.28	15.15	0.06%	1111.38	9.31	0.15%
	189	100	990	4896	829	1110	1109.32	10.67	0.06%	1109.31	15.26	0.06%	1108.93	9.56	0.10%
	190	100	990	4896	835	1090	1088.60	10.73	0.13%	1088.45	15.10	0.14%	1087.81	9.41	0.20%
	191	100	990	19583	841	-	1166.80	14.32		1136.67	32.57		1171.31	10.10	
	192	100	990	19583	847	1491*	1133.20	14.21	24.00%	1101.82	34.00	26.10%	1136.95	10.04	23.75%
	193	100	990	19583	853	1510*	1131.15	14.11	25.09%	1098.78	34.71	27.23%	1138.10	10.07	24.63%
	194	100	990	19583	859	1441*	1168.84	14.18	18.89%	1140.83	35.59	20.83%	1175.92	10.06	18.40%
	195	100	990	19583	865	1560*	1167.81	14.23	25.14%	1137.83	34.19	27.06%	1172.95	10.10	24.81%
	196	100	990	34269	871	-	1110.46	20.01		1086.82	39.51		1124.06	10.97	
	197	100	990	34269	877	-	1133.88	20.02		1114.14	35.35		1147.24	10.93	
	198	100	990	34269	883	-	1164.32	20.11		1139.40	37.06		1176.89	11.03	
	199	100	990	34269	889	-	1131.12	20.24		1107.71	38.19		1144.40	10.97	
	200	100	990	34269	895	-	1138.86	20.80		1118.75	38.69		1153.32	11.06	
	201	100	1485	11019	901	1079	1077.48	31.35	0.14%	1076.82	38.85	0.20%	1076.36	28.98	0.24%
	202	100	1485	11019	907	1056	1054.41	31.38	0.15%	1054.31	42.26	0.16%	1054.02	29.01	0.19%
	203	100	1485	11019	913	1059	1058.87	31.38	0.01%	1058.49	38.41	0.05%	1058.55	29.12	0.04%
	204	100	1485	11019	919	1046	1045.91	31.40	0.01%	1044.28	38.94	0.16%	1045.99	29.59	0.00%
	205	100	1485	11019	925	1072	1070.88	31.41	0.10%	1071.12	37.41	0.08%	1070.70	28.86	0.12%
	206	100	1485	44075	931	1374*	1088.31	44.82	20.79%	1080.23	93.18	21.38%	1092.12	33.10	20.52%
	207	100	1485	44075	937	1291*	1082.14	44.63	16.18%	1071.94	90.21	16.97%	1087.75	33.51	15.74%
	208	100	1485	44075	943	1344*	1076.47	44.56	19.91%	1065.54	91.77	20.72%	1080.36	33.37	19.62%
	209	100	1485	44075	949	1286*	1079.84	44.62	16.03%	1070.08	94.00	16.79%	1085.76	34.23	15.57%
	210	100	1485	44075	955	1370*	1079.56	44.55	21.20%	1070.28	90.39	21.88%	1085.31	33.77	20.78%
	211	100	1485	77131	961	-	1068.27	57.99		1064.12	108.55		1081.62	34.91	
	212	100	1485	77131	967	-	1070.55	57.93		1067.31	114.58		1084.68	34.96	
	213	100	1485	77131	973	-	1078.82	58.02		1075.46	108.57		1091.32	35.02	
	214	100	1485	77131	979	-	1090.95	57.76		1089.89	110.82		1104.36	35.07	
	215	100	1485	77131	985	-	1076.89	57.25		1073.70	113.32		1088.92	35.52	
	216	100	1980	19593	991	1031	1030.37	68.96	0.06%	1030.15	91.28	0.08%	1029.92	65.54	0.10%
	217	100	1980	19593	997	1036	1034.39	68.95	0.16%	1034.49	95.75	0.15%	1034.22	65.31	0.17%
	218	100	1980	19593	1003	1024	1023.85	68.68	0.01%	1023.05	88.93	0.09%	1023.84	65.24	0.02%
	219	100	1980	19593	1009	1025	1024.99	68.78	0.00%	1024.06	82.94	0.09%	1024.98	65.21	0.00%
	220	100	1980	19593	1015	1028	1027.33	68.93	0.07%	1026.71	90.78	0.13%	1026.94	65.40	0.10%
	221	100	1980	78369	1021	1234*	1050.68	97.54	14.86%	1050.21	183.40	14.89%	1059.58	85.90	14.13%
	222	100	1980	78369	1027	1187*	1024.58	97.33	13.68%	1023.72	189.59	13.76%	1031.56	83.03	13.10%
	223	100	1980	78369	1033	1213*	1042.67	97.57	14.04%	1048.06	181.54	13.60%	1048.52	79.67	13.56%
	224	100	1980	78369	1039	1221*	1039.69	97.53	14.85%	1045.69	184.63	14.36%	1045.12	85.89	14.40%
	225	100	1980	78369	1045	1245*	1040.94	97.58	16.39%	1040.26	186.59	16.44%	1045.92	82.74	15.99%
	226	100	1980	137145	1051	-	1028.22	120.89		1032.43	208.32		1039.17	80.01	
	227	100	1980	137145	1057	-	1051.28	120.74		1056.49	205.90		1064.84	79.60	
	228	100	1980	137145	1063	-	1038.61	120.78		1043.29	219.03		1050.80	79.54	
	229	100	1980	137145	1069	-	1048.04	120.42		1054.00	204.37		1058.74	79.42	
	230	100	1980	137145	1075	-	1040.94	120.66		1045.37	231.09		1054.43	79.35	
AVG								**31.16**	**7.17%**		**55.09**	**7.60%**		**23.48**	**7.01%**

Table 7.3: Comparison of three algorithms on the large instances proposed in [11].

573–604, 2017.

[17] Frangioni, A., B. Gendron and E. Gorgone, Dynamic smoothness parameter for fast gradient methods, *Optimization Letters* 12: 43–53, 2018.

[18] Fumero, F., A modified subgradient algorithm for lagrangean relaxation, *Computers and Operations Research* 28(1):33–52, 2001.

[19] Gaudioso, M., G. Giallombardo, G. Miglionico, Essentials of numerical nonsmooth optimization, *4OR*, 18, 1: 1–47, 2020.

[20] Gaudioso, M., A view of Lagrangian relaxation and its applications. In *Numerical nonsmooth optimization–State of the art algorithms*, Bagirov, A.M, M. Gaudioso, N. Karmitsa, M. Mäkelä and S. Taheri, eds., Springer, 2020.

[21] Geoffrion, A., Lagrangean relaxation and its uses in integer programming, *Mathematical Programming Study* 2:82–114, 1974.

[22] Hiriart-Urruty, J.-B., and C. Lemaréchal, Convex Analysis and Minimization Algorithms, Vol. I and II, Springer-Verlag, Berlin– Heidelberg, 1993.

[23] Kiwiel, K. C., Proximity control in bundle methods for convex nondifferentiable minimization, *Mathematical Programming* 46: 105–122, 1990.

[24] López-Ibáñez, M., J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, T. Stützle, The Irace package: Iterated racing for automatic algorithm configuration, *Operations Research Perspectives*, 3:43-58, 2016.

[25] Lemaréchal, C., An algorithm for minimizing convex functions, in J.L. Rosenfeld, ed., *Proceedings IFIP '74 Congress*, 20–25, North-Holland, Amsterdam, 1974.

[26] Lemaréchal, C. and R. Mifflin eds., Nonsmooth Optimization, 1978, Pergamon Press, Oxford.

[27] Lukšan, L, J. Vlček, Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, Tech. Rep. 798, Institute of Comp. Sc., Academy of Sciences of the Czech Republic, 2000.

[28] Nemirovski, A., D. Yudin, Problem complexity and method efficiency in optimization, Wiley, New York, 1983.

[29] Nesterov, Yu., Smooth minimization of non-smooth functions, *Mathematical Programmming* 103: 127–152, 2005.

[30] Nesterov, Yu., Primal-dual subgradient methods for convex problems, *Mathematical Programmming* 120: 221–259, 2009.

[31] Nesterov, Yu., Universal gradient methods for convex optimization problems, *Mathematical Programmming* 152: 381–404, 2009.

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

[33] Rockafellar, R.T., Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization* 14: 877–898, 1976.

[34] P. Samer and S. Urrutia, A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters* 9(1):41–55, 2014.

[35] Samer, P., D. Haugland, Polyhedral results and stronger Lagrangean bounds for stable spanning trees. it Optimization Letters, 2022. https://doi.org/10.1007/s11590-022-01949-8

[36] Shor, N.Z., Minimization methods for nondifferentiable functions, Springer-Verlag, Berlin, 1985.

[37] Zhang, R., S.N. Kabadi, and A.P. Punnen, The minimum spanning tree problem with conflict constraints and its variations. it Discrete Optimization, 8(2):191–205, 2011.