**Mathematical Programming manuscript No.**
(will be inserted by the editor)

L. Bahiense · N. Maculan · C. Sagastizábal

# The Volume Algorithm revisited: relation with bundle methods.

**Abstract.** We revise the Volume Algorithm (VA) for linear programming and relate it to bundle methods. When first introduced, VA was presented as a subgradient-like method for solving the original problem in its dual form. In a way similar to the serious/null steps philosophy of bundle methods, VA produces green, yellow or red steps. In order to give convergence results, we introduce in VA a precise measure for the improvement needed to declare a green or serious step. This addition yields a revised formulation (RVA) that is halfway between VA and a specific bundle method, that we call BVA. We analyze the convergence properties of both RVA and BVA. Finally, we compare the performance of the modified algorithms versus VA on a set of Rectilinear Steiner problems of various sizes and increasing complexity, derived from real world VLSI design instances.

**Key words.** volume algorithm – bundle methods – Steiner problems

## 1. Introduction

Consider the problem of solving large-scale linear programs of the form

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} \ \langle c, x\rangle & \\ A\,x = b & \text{(a)} \\ D\,x = e & \\ x \geq 0\,, & \end{cases} \tag{1}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $D \in \mathbb{R}^{d\times n}$, $e \in \mathbb{R}^d$, and $rank\, D = d$. When constraints (1)(a) are difficult to deal with, a possible approach is to solve a dual problem, obtained via Lagrangian relaxation. For this approach to be efficient, two critical points are:

– how to solve the nondifferentiable dual problem,

– how to recover a primal solution.

The *Volume Algorithm* (VA) introduced in [2] appears as a good answer to the questions above:

Universidade Federal do Rio de Janeiro, COPPE-Sistemas e Computacão, P.O.Box 68511, Rio de Janeiro, RJ 21945-970, Brazil. The third author is also with INRIA- Rocquencourt, BP 105, 78153 Le Chesnay, France. e-mail: {`laura,maculan,sagastiz`}`@cos.ufrj.br`

*Correspondence to*: Claudia A. Sagastizábal. Current address: IMPA, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro RJ 22460-320, Brazil. E-mail: sagastiz@impa.br

– a dual optimum is obtained by applying what the authors call an *extension* of the subgradient algorithm, [16]. Such extension aims at keeping the simplicity of the subgradient methods while mimicking a bundle-like strategy, in the spirit of the very first works [14] and [21], i.e., with quadratic subproblems that are easy to solve, because they are bivariate;
– a primal solution is simultaneously produced by estimating certain volumes associated to active faces in (1)(a).

To assess the validity of VA, in [2, Section 6] successful numerical experience is reported on set partitioning, set covering, max-cut and facility location problems.

Actually, [2] mostly stresses numerical concerns, and does not consider theoretical properties of VA, such as when the proposed algorithm converges, or its relation with other methods. In this paper we address this issue and show that VA can be interpreted as an extragradient method, [13], [17]. More precisely, each iteration of VA generates a sampling point by using an $\varepsilon$-extragradient, see Theorem 1 below. A sampling point is declared to be *green* in [2] when there is an improvement in the dual function, somewhat similarly to a serious-step in bundle methods. However, unlike bundle methods, in VA the objective improvement is not rigorously measured.

In order to keep track of the improvement produced by each green or serious step, it is important to supply VA with the notion of a *model* of the objective dual function $\theta$. As a first step towards analyzing convergence, we present a revised variant, that we call RVA. This revised formulation introduces a precise measure for the improvement needed to declare a green iteration. Such measure, known also as *expected gain*, relates exact values of $\theta$ to values predicted by a computable function modelling $\theta$. We conclude our analysis by relating RVA to BVA, an economic variant of bundle methods.

Our paper is organized as follows. In Section 2 we revise some essential notions of duality and convexity. Next, in Section 3, we present VA and study some of its main features. All the elements needed to introduce a model in VA are developed in Section 4. Section 5 is devoted to RVA: description and convergence analysis. When RVA generates an infinite sequence of null steps, we only obtain a convergence result that depends on a rather strong assumption, namely (40) below. In order to give a full convergence analysis, we are bound to further modify RVA and mold it as a bundle method, yielding BVA. This is the subject of Section 6, where we also give an a-posteriori error bound for the approximated primal solution obtained from RVA. Finally, we report in Section 7 successful numerical results comparing RVA and and BVA with VA on a battery of rectilinear Steiner problems of various sizes, [12] and [3].

## 2. A glimpse of duality and convexity

We first review some basic notions of classical Lagrangian relaxation and convex(concave) analysis, [4, Ch. VIII]. In particular, in Section 2.2, we recall the important concept of $\varepsilon$-supergradient.

### 2.1. *The dual problem*

In order to relax constraints (1)(a) we use the Lagrangian function

$$L(x,\pi) := \langle c, x\rangle + \langle Ax - b, \pi\rangle\,, \tag{2}$$

where $\pi \in \mathbb{R}^m$ is the associated vector of Lagrange multipliers.

We suppose that (1) is feasible. In addition, we denote by $\Psi$ the nonempty polyhedron gathering the "easy" constraints:

$$\Psi := \{x \in \mathbb{R}^n : Dx = e\,,\ x \geq 0\,\}\,. \tag{3}$$

In this context, the weak duality relationship

$$\min_{x\in\Psi}\max_{\pi\in\mathbb{R}^m} L(x,\pi) \geq \max_{\pi\in\mathbb{R}^m}\min_{x\in\Psi} L(x,\pi) \tag{4}$$

holds. The left-hand side of (4), or *primal* problem, has the same optimal value as (1). The right-hand side in (4) is the *dual* problem

$$\max_{\pi\in\mathbb{R}^m} \theta(\pi)\,, \tag{5}$$

whose objective is the *dual* function

$$\theta(\pi) := \min_{x\in\Psi} L(x,\pi)\,. \tag{6}$$

Equality in (4) means that there is no duality gap. That is to say, solving (5) amounts to solve (1). For our problem, given a point $p^\star \in \mathcal{P}^\star$, the solution set of (5), any $x(p^\star)$ satisfying

$$x(p^\star) \in \operatorname*{Argmin}_{x\in\Psi} L(x,p^\star) \quad \text{ such that } Ax(p^\star) = b$$

is a solution of (1).

### 2.2. *Supergradients and $\varepsilon$-supergradients*

Being defined as the pointwise minimum of affine functions of $\pi$, the dual function (6) is concave and nondifferentiable. To maximize $\theta$, it is known that any nonsmooth optimization method uses the information given by an "oracle". Specifically, for any given $\pi$, the oracle returns the values $\theta(\pi)$ and a subgradient of the convex function $-\theta$ at $\pi$. Since throughout this paper we refer to the concave function $\theta$, we find convenient to introduce here the notion of *super*gradients.

**Definition 1.** Let $\theta$ be a concave function and let $\pi \in \mathbb{R}^m$. The point $v \in \mathbb{R}^m$ is called a *supergradient* of $\theta$ at $\pi$ whenever

$$\theta(\pi') \leq \theta(\pi) + \langle v, \pi' - \pi\rangle \ \text{ for all } \pi' \in \mathbb{R}^m. \tag{7}$$

In this case, we write $v \in \partial\theta(\pi)$.
Given $\varepsilon \geq 0$, the point $w \in \mathbb{R}^m$ is an *$\varepsilon$-supergradient* of $\theta$ at $p \in \mathbb{R}^m$ whenever

$$\theta(\pi') \leq \theta(p) + \langle w, \pi' - p\rangle + \varepsilon \ \text{ for all } \pi' \in \mathbb{R}^m\,. \tag{8}$$

In this case, we write $w \in \partial\theta_\varepsilon(p)$. □

The $\varepsilon$-superdifferential $\partial_\varepsilon \theta(p)$ has good continuity properties, see [7, XI.4.1.1]. In particular, it is a closed multi-function of $\varepsilon$ and $p$:

$$\{\varepsilon_t\,, p_t\,, w_t \in \partial_{\varepsilon_t}\theta(p_t)\} \longrightarrow \{\varepsilon^*\,, p^*\,, w^*\} \Longrightarrow w^* \in \partial_{\varepsilon^*}\theta(p^*)\,. \tag{9}$$

When specialized to our dual function (6), we see that to evaluate $\theta(\pi)$ the oracle must solve the subproblem

$$\theta(\pi) = \min_{x\in\Psi} L(x,\pi) = \min_{x\in\Psi}\{\langle c, x\rangle + \langle Ax - b, \pi\rangle\}\,. \tag{10}$$

Let $x(\pi) \in \operatorname{Argmin} L(\cdot,\pi)$ be a solution of (10). Then, for any $\pi' \in \mathbb{R}^m$, straightforward calculations yield

$$\begin{aligned}\theta(\pi') &= \min_{x\in\Psi} L(x,\pi') \leq L(x(\pi),\pi')\langle c, x(\pi)\rangle + \langle Ax(\pi) - b, \pi'\rangle \\ &= \langle c, x(\pi)\rangle + \langle Ax(\pi) - b, \pi'\rangle \pm \langle Ax(\pi) - b, \pi\rangle \\ &= \theta(\pi) + \langle Ax(\pi) - b, \pi' - \pi\rangle\,.\end{aligned}$$

It follows that the supergradient

$$v := Ax(\pi) - b \in \partial\theta(\pi) \tag{11}$$

can be computed for free once the oracle has solved (10). Clearly, the difficulty for solving (10) depends on the nature of the constraints defining the polyhedron $\Psi$ of (3). The easier this subproblem solution will be, the more efficient will be the dual approach. This is precisely the case of Steiner problems considered in our numerical experience, see Section 7.2 below.

## 3. On the volume method

The dual problem (5) needs to be solved by using a nonsmooth optimization method, such as subgradients, cutting-planes, analytic centers, or bundle methods, see [7]. All of these methods have advantages and drawbacks, and their efficiency depends on the nature of the problem to be solved. For instance, subgradient methods are known by their simplicity, but also for the lack of well defined stopping criteria. On the other hand, bundle methods are known to be robust and precise, but at each iteration the solution of a (potentially heavy) quadratic program is required.

The volume methodology was introduced in [2] in an effort to combine the best features of subgradients and bundle methods. Roughly speaking, to produce a dual solution VA generates sampling points $\pi_t$ by solving a subproblem as in (10), with $\pi_t$ depending on:

– a given stability center $\hat{\pi}_k$,
– a certain steplength $s_t$, and
– $w_t$, a convex combination of available supergradients.

Stability centers are special sampling points, providing a "good enough" improvement in the optimization process; in other words, $\{\hat{\pi}_k\}$ is a selected subsequence of $\{\pi_t\}$. Primal solutions are approximated by $z_t$, a convex combination of past primal points. The coefficients used in such convex sums are the same than those used to compute $w_t$, see (14), (15) below.

### *3.1. The Volume Algorithm VA*

STEP 0. Given an initial $\pi_0 \in \mathbb{R}^m$, compute $x_0 \in \operatorname{Argmin} L(x, \pi_0)$, a solution of (10) written for $\pi = \pi_0$. Let $v_0 := Ax_0 - b$. Initialize $z_1 = x_0$, $w_1 := v_0$, and $\hat{\pi}_1 = \pi_0$. Set $k = t = 1$ and $T_s = \emptyset$.

STEP 1. Having the stability center $\hat{\pi}_k$, and asteplength $s_t > 0$, make the move

$$\pi_t = \hat{\pi}_k + s_t w_t \,. \tag{12}$$

STEP 2. Depending on whether

$$\theta(\pi_t) > \theta(\hat{\pi}_k) \qquad \text{and} \qquad \langle w_t, v_t \rangle > 0 \tag{13}$$

is false or true, decide to make, respectively,

– either a *null-step*: (13) does not hold, just do nothing,

– or a *serious-step*: (13) holds. In this case, update the stability center: $\hat{\pi}_{k+1} = \pi_t$, set $t_k := t$, $T_s = T_s \cup \{t_k\}$, and increase $k$ by 1.

STEP 3. Compute $x_t \in \operatorname{Argmin} L(x, \pi_t)$, a solution of (10) written for $\pi = \pi_t$. Let $v_t := Ax_t - b$.

STEP 4. Compute a new stepsize $s_{t+1}$.

Given a parameter $0 \leq \alpha_t \leq 1$, define

$$z_{t+1} := \alpha_t x_t + (1 - \alpha_t) z_t \tag{14}$$

$$w_{t+1} := \alpha_t v_t + (1 - \alpha_t) w_t \,. \tag{15}$$

This completes the $t^{th}$ iteration: set $t = t + 1$ and loop to 1. □

When comparing our description with the original statement of VA in [2] a few minor differences arise:

- Serious steps correspond to *green* iterations in the Volume Algorithm, but we have collapsed *yellow* and *red* iterations in our null steps.
- The original Volume Algorithm further specifies the stepsize as

$$s_t = \mu \, \frac{\text{UB} - \theta(\pi_t)}{\|w_t\|^2} \,, \tag{16}$$

  where UB is an upper bound for the optimal value in (1) and $\mu \in (0, 2)$ is a relaxation factor. Since our development does not rely on this particular choice of steplength, we prefer to omit it in our description of VA. Instead, for our convergence results, we will establish abstract conditions to be satisfied by the sequence of stepsizes, such as (35), (37) and (41) given in Section 5.2.
- A suitable choice of the convex parameter $\alpha_t$ is discussed in Section 4 below.

### *3.2. Some useful relations*

We now establish some relations satisfied by the recurrent formulæ of VA.

**Lemma 1.** *Let $\{z_t\}$, $\{v_t\}$ and $\{w_t\}$ be the sequences generated by VA. Then for all $t$, $v_t \in \partial\theta(\pi_t)$ and $w_t = Az_t - b$.*

*Proof.* The first statement is just (11), written with $(\pi, v, x(\pi)) := (\pi_t, v_t, x_t)$. The second one is straightforward from the definition of each sequence involved, using the linearity of $A$. □

Since $v_t \in \partial\theta(\pi_t)$ for all $t$, from (7) it can be seen that the righthand side condition in (13) implies the lefthand side condition in the same equation. We conjecture that this was the reason behind the distinction between *yellow* and *red* iterations in the original Volume Algorithm.

**Lemma 2.** *For any $t \geq 1$, consider the consider the following coefficients*

$$\mu_{t,j} := \alpha_{t-j} \prod_{i=t-j+1}^{t} (1-\alpha_i) \quad \text{for } j = 0, \ldots, t\,, \tag{17}$$

*where $\alpha_0 := 1$ and the product $\prod_{i=i_0}^{i_f}(1-\alpha_i)$ is defined to be 1 whenever $i_f < i_0$. The following hold:*

*(i) for all $j \leq t$, $\mu_{t,j} \geq 0$, and $\sum_{j=0}^{t} \mu_{t,j} = 1$; and*
*(ii) both (14) and (15) can be expressed using the convex multipliers $\mu_{t,j}$:*

$$z_{t+1} = \sum_{j=0}^{t} \mu_{t,j} x_{t-j} \qquad \text{and} \qquad w_{t+1} = \sum_{j=0}^{t} \mu_{t,j} v_{t-j}\,.$$

*Proof.* [(i)] Since $\alpha_t \in [0,1]$ for all $t$, positivity of $\mu_{t,j}$ is clear. To see that the coefficients sum up to 1, expand the terms of the sum taking nested factors:

$$\begin{aligned}\sum_{j=0}^{t} \mu_{t,j} &= \alpha_t 1 + \alpha_{t-1}(1-\alpha_t) + \alpha_{t-2}(1-\alpha_t)(1-\alpha_{t-1}) + \cdots \\ &= \alpha_t + (1-\alpha_t)[\alpha_{t-1} + (1-\alpha_{t-1})(\alpha_{t-2} + (1-\alpha_{t-2})(\cdots(1-\alpha_2)(\alpha_1 + 1 - \alpha_1)\cdots)] \\ &= \alpha_t + (1-\alpha_t)[\alpha_{t-1} + (1-\alpha_{t-1})(\alpha_{t-2} + (1-\alpha_{t-2})1)] \\ &= \alpha_t + (1-\alpha_t)[\alpha_{t-1} + (1-\alpha_{t-1})1] \\ &= 1\,.\end{aligned}$$

The proof of (ii) follows from (14) (resp. (15)), by induction on $t$. □

We now show that each $w_t$ is an approximate supergradient of $\theta$ at a point $p_t$, that can be defined recursively using a formula like the ones in (14), (15). Namely, letting $p_1 := \pi_0$, at Step 4 of VA we also compute

$$p_{t+1} := \alpha_t \pi_t + (1-\alpha_t) p_t\,. \tag{18}$$

An equivalent expression, obtained by reasoning like in Lemma 2$(ii)$, is

$$p_{t+1} = \sum_{j=0}^{t} \mu_{t,j} \pi_{t-j} \,. \tag{19}$$

**Theorem 1.** *Let* $\{z_t\}$, $\{w_t\}$ *and* $\{p_t\}$ *be the sequences generated by* (14), (15) *and* (18), *respectively. For any* $t \geq 1$, *consider the coefficients* $\{\varepsilon_t\}$ *defined as*

$$\varepsilon_{t+1} := \alpha_t(1-\alpha_t)\langle v_t - w_t, p_t - \pi_t\rangle + (1-\alpha_t)\varepsilon_t \,, \tag{20}$$

*with* $\varepsilon_1 := 0$. *Then* $\varepsilon_t \geq 0$ *and* $w_t \in \partial_{\varepsilon_t}\theta(p_t)$.

*Proof.* When $t = 1$, $w_1 = v_0 = Ax_0 - b \in \partial\theta(\pi_0)$ by Lemma 1. Since $p_1 = \pi_0$ and $\varepsilon_1 = 0$, $w_1 \in \partial_{\varepsilon_1}\theta(p_1) = \partial\theta(\pi_0)$.
For all $t \geq 1$, make the inductive assumption that $w_t \in \partial_{\varepsilon_t}\theta(p_t)$. Since $v_t \in \partial\theta(\pi_t)$ by Lemma 1, using (7) and (8), we have for all $\pi' \in \mathbb{R}^m$:

$$\begin{array}{ll} \theta(\pi') \leq \theta(\pi_t) + \langle v_t, \pi' - \pi_t\rangle & \text{(a)} \\ \theta(\pi') \leq \theta(p_t) + \langle w_t, \pi' - p_t\rangle + \varepsilon_t & \text{(b)}. \end{array}$$

Make the convex combination $\alpha_t$(a)+$(1-\alpha_t)$(b), use (18) and the concavity of $\theta$ to write, for all $\pi' \in \mathbb{R}^m$,

$$\theta(\pi') \;\leq\; \theta(p_{t+1}) + \langle w_{t+1}, \pi'\rangle - \langle \alpha_t v_t, \pi_t\rangle - \langle (1-\alpha_t)w_t, p_t\rangle + (1-\alpha_t)\varepsilon_t \,.$$

Call $\varepsilon_{t+1}$ the term such that the inequality above can be written as

$$\theta(\pi') \;\leq\; \theta(p_{t+1}) + \langle w_{t+1}, \pi' - p_{t+1}\rangle + \varepsilon_{t+1} \,,$$

$$\begin{aligned} \text{i.e.,}\ \ \varepsilon_{t+1} &:= \langle w_{t+1}, p_{t+1}\rangle - \langle \alpha_t v_t, \pi_t\rangle - \langle (1-\alpha_t)w_t, p_t\rangle + (1-\alpha_t)\varepsilon_t \\ &= \langle \alpha_t v_t, p_{t+1} - \pi_t\rangle + \langle (1-\alpha_t)w_t, p_{t+1} - p_t\rangle + (1-\alpha_t)\varepsilon_t \,. \end{aligned}$$

By (18), $p_{t+1} - \pi_t = (1-\alpha_t)\,(p_t - \pi_t)$ and $p_{t+1} - p_t = \alpha_t\,(\pi_t - p_t)$. Thus,

$$\begin{aligned} \varepsilon_{t+1} &= \langle \alpha_t v_t, (1-\alpha_t)\,(p_t - \pi_t)\rangle + \langle (1-\alpha_t)w_t, \alpha_t\,(\pi_t - p_t)\rangle + (1-\alpha_t)\varepsilon_t \\ &= \alpha_t(1-\alpha_t)\langle v_t - w_t, p_t - \pi_t\rangle + (1-\alpha_t)\varepsilon_t \,. \end{aligned}$$

Finally, apply (7) with $(\pi', \pi, v) = (p_t, \pi_t, v_t)$ and (8) with $(\pi', p, \varepsilon, w) = (\pi_t, p_t, \varepsilon_t, w_t)$, respectively, to see that $\langle v_t - w_t, \pi_t - p_t\rangle \leq \varepsilon_t$ Altogether, we obtain

$$\varepsilon_{t+1} \geq -\alpha_t(1-\alpha_t)\varepsilon_t + (1-\alpha_t)\varepsilon_t = (1-\alpha_t)^2\,\varepsilon_t \,,$$

a nonnegative quantity, by construction. □

The sequence $\{\varepsilon_t\}$ is defined by the recursion (20), analogous to those defining $\{z_t\}$, $\{w_t\}$ and $\{p_t\}$. As a result, a formula along the lines of the ones in Lemma 2$(ii)$ and (19) also holds for $\{\varepsilon_t\}$:

$$\varepsilon_{t+1} = \sum_{j=0}^{t} \mu_{t,j}\sigma_{t-j} \quad \text{for } \sigma_t := (1-\alpha_t)\langle v_t - w_t, p_t - \pi_t\rangle \,. \tag{21}$$

Theorem 1 shows that STEP 1 in VA uses a supergradient that is not computed at the current iterate $\hat{\pi}_k$, but at a *different* point, namely $p_t$. In this respect, VA is closer to the *extragradient* approach [13], than to a subgradient method. Note also that, since $w_t \in \partial_{\varepsilon_t}\theta(p_t)$, the extragradient used by VA is only an approximate one.

Although the "inexact" move (12) is not the most common extragradient update, it is not the most unusual either. There are other methods that do use "inexacteness" both in the point where the gradient is computed and in the gradient itself (i.e., $\varepsilon_t > 0$). For example, the modified forward-backward splitting method of [19], that can be regarded as an implementation of the approximate extragradient proximal method for generalized equations, see [17, Section 5].

In addition, note that, when compared to a typical bundle iteration, the crucial distinction between serious- and null-steps is not kept with precision, since (13) in VA does not measure the gain obtained. Specifically, when passing from $\hat{\pi}_k$ to $\hat{\pi}_{k+1}$, it is not known how much "better" the serious step is, we only know that $\theta(\hat{\pi}_{k+1})$ is bigger than $\theta(\hat{\pi}_k)$.

We address this point in the next section, by introducing the concept of expected gain in a revised version of VA.

## 4. Towards a convergent method

In order to keep track of the improvement produced by each serious step, it is important to replace (13) by a condition of the type $\theta(\pi_t) \geq \theta(\hat{\pi}_k) + m_1\delta_t$, where $m_1$ is an Armijo-like tolerance and, more importantly, $\delta_t > 0$ measures the *expected gain*, associated to a computable function modelling $\theta$.

### *4.1. Introducing a model*

Typically, bundle methods make use of a *model* $\hat{\theta}$ to approximate the unknown function $\theta$. The modelling concave function varies along iterations and is defined as the minimum of planes tangent to *graph* $\theta$. Given $\pi_j$, a call to the oracle (i.e., solving (10)) gives $\theta(\pi_j)$ and $v_j \in \partial\theta(\pi_j)$. The corresponding cutting plane is

$$\theta(\pi_j) + \langle v_j, \pi - \pi_j \rangle .$$

An equivalent expression, more convenient for our development, refers this plane to the last serious-step $\hat{\pi}_k$:

$$\theta(\hat{\pi}_k) + \langle v_j, \pi - \hat{\pi}_k \rangle + e_j \qquad \text{with } e_j := \theta(\pi_j) + \langle v_j, \hat{\pi}_k - \pi_j \rangle - \theta(\hat{\pi}_k) .$$

The model is defined by the function value at the last serious-step and by a given number of cutting planes. Here we use an economic bundle of information, containing only two cutting planes: first, the plane generated by the last $\pi_t$, and, second, the so-called *aggregate* plane. More precisely, suppose that at STEP 4 in VA the following (non negative) quantities are available:

$$e_t := \theta(\pi_t) + \langle v_t, \hat{\pi}_k - \pi_t \rangle - \theta(\hat{\pi}_k) \quad \text{and} \tag{22}$$

$$\hat{e}_t := \theta(p_t) + \langle w_t, \hat{\pi}_k - p_t \rangle + \varepsilon_t - \theta(\hat{\pi}_k)\,. \tag{23}$$

The corresponding economic model $\hat{\theta}_t$ has the form

$$\hat{\theta}_t(\pi) := \theta(\hat{\pi}_k) + \min\{e_t + \langle v_t, \pi - \hat{\pi}_k \rangle\,, \hat{e}_t + \langle w_t, \pi - \hat{\pi}_k \rangle\}\,. \tag{24}$$

By concavity, the hyperplane $\mathcal{H} := \theta(\hat{\pi}_k) + \{\pi \in \mathbb{R}^m : \langle v_t, \pi - \hat{\pi}_k \rangle + e_t = 0\}$ supports (from above) the graph of $\theta$ at $\pi_t$. Likewise, the hyperplane $\mathcal{H}_\varepsilon := \theta(\hat{\pi}_k) + \{\pi \in \mathbb{R}^m : \langle w_t, \pi - \hat{\pi}_k \rangle + \hat{e}_t = 0\}$ supports within $\varepsilon_t$ the graph of $\theta$ at $p_t$. Then, as shown in Figure 1, the function $\hat{\theta}_t$ from (24) is a cutting-planes model

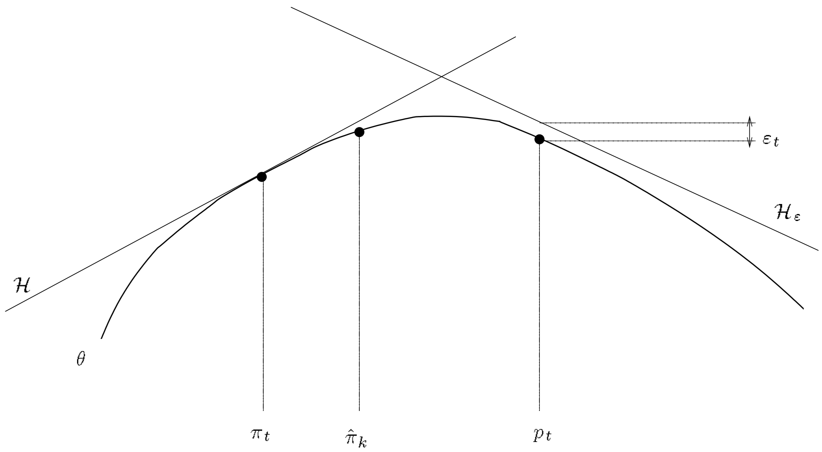


**Fig. 1.** Model $\hat{\theta}$.

for $\theta$ such that $\hat{\theta}_t(\pi) \geq \theta(\pi)$ for all $\pi \in \mathbb{R}^m$.

The model $\hat{\theta}_t$ is used to compute the next iterate, say $\pi_{t+1}$. The expected gain $\delta_t$, relating exact value functions to approximate ones predicted by the model is usually defined as $\delta_t := \hat{\theta}_t(\pi_{t+1}) - \theta(\hat{\pi}_k)$. In view of (24), this yields

$$\delta_{t+1} := \min\{e_t + \langle v_t, \pi_{t+1} - \hat{\pi}_k \rangle, \hat{e}_t + \langle w_t, \pi_{t+1} - \hat{\pi}_k \rangle\}\,. \tag{25}$$

A more concise form for $\delta_t$ makes use of the parameters $\alpha_t$, see (29) below.

### *4.2. Choosing $\alpha_t$*

So far, nothing has been said on how to choose $\alpha_t$ in STEP 4 in VA. We now show how to make a sound choice of these parameters, if a model $\hat{\theta}_t$ is available.

More precisely, consider the (strongly convex) problem

$$\max_{\pi \in \mathbb{R}^m} \hat{\theta}_t(\pi) - \frac{1}{2s_{t+1}} \|\pi - \hat{\pi}_k\|^2\,, \tag{26}$$

whose unique solution $\pi'$ is characterized by the optimality condition

$$\exists w' \in \partial\hat{\theta}_t(\pi') \text{ such that } w' - \frac{1}{s_{t+1}}(\pi' - \hat{\pi}_k) = 0 \quad [\text{i.e., } \pi' = \hat{\pi}_k + s_{t+1}w'.]$$

Since $\hat{\theta}_t$ is the maximum of two affine functions, its superdifferential is $\partial\hat{\theta}_t(\pi) = \{\alpha v_t + (1-\alpha)w_t : \alpha \in [0,1]\}$ for all $\pi$. Thus, the optimality condition becomes

$$\exists \alpha' \in [0,1] \text{ such that } \pi' = \hat{\pi}_k + s_{t+1}w' \text{ with } w' = \alpha' v_t + (1-\alpha')w_t\,.$$

As a result, choosing in STEP 4 of VA $\alpha_t := \alpha'$ implies that the gradient $w'$ given by the optimality condition is precisely $w_{t+1}$ from (15). Furthermore, with this choice of $\alpha_t$ the next iterate $\pi_{t+1}$ from (12) will be precisely the unique point $\pi'$ given by the optimality condition above:

$$\exists \alpha_t \in [0,1] \text{ such that } w_{t+1} \in \partial\hat{\theta}_t(\pi_{t+1}) \text{ and } \pi_{t+1} = \hat{\pi}_k + s_{t+1}w_{t+1}\,. \tag{27}$$

The convex parameter $\alpha_t$ can also be found by solving a problem dual to (26):

$$\min_{\alpha\in[0,1]} \frac{s_{t+1}}{2}\|\alpha v_t + (1-\alpha)w_t\|^2 + \alpha e_t + (1-\alpha)\hat{e}_t\,. \tag{28}$$

This choice of $\alpha_t$ yields the following expression for the expected gain:

$$\delta_{t+1} = s_{t+1}\|w_{t+1}\|^2 + \alpha_t e_t + (1-\alpha_t)\hat{e}_t\,. \tag{29}$$

### *4.3. Supplying the volume method with a model*

When trying to incorporate the notion of a model in VA, an important question arises. Namely, to define $\hat{e}_t$ in (23), one needs to know $\theta(p_t)$, a value that is not computed in VA. The only available functional values are $\theta(\pi_t)$, computed at STEP 3. To address this issue, we consider two possibilities:

- a "minimalistic" approach, in which we keep as close as possible to VA and, instead of computing the extra value function $\theta(p_t)$, we approximate $\hat{e}_t$ with quantities that are already available.
- an "everything-changes" approach, in which a reorganization of calculations allows us to define $\hat{e}_t$ without using $p_t$.

The first approach results in the revised volume algorithm RVA, while the second is BVA, the economic variant of bundle methods. We analyze here the first variant, and defer the analysis of the second one to Section 6.2.

To approximate the unknown $\theta(p_t)$ we use in (23) the best available functional value, i.e., $\theta(\hat{\pi}_k)$ :

$$\hat{e}_t = \theta(p_t) + \langle w_t, \hat{\pi}_k - p_t\rangle + \varepsilon_t - \theta(\hat{\pi}_k) \approx \langle w_t, \hat{\pi}_k - p_t\rangle\,,$$

and similarly in (22), for the sake of consistency. Hence, we use the quantities

$$E_t := \langle v_t, \hat{\pi}_k - \pi_t\rangle \quad \text{and} \tag{30}$$

$$\hat{E}_t := \langle w_t, \hat{\pi}_k - p_t\rangle + \varepsilon_t\,. \tag{31}$$

The resulting model $\hat{\Theta}_t(\pi) := \theta(\hat{\pi}_k) + \min\{E_t + \langle v_t, \pi - \hat{\pi}_k\rangle\, , \hat{E}_t + \langle w_t, \pi - \hat{\pi}_k\rangle\}$ is shown in Figure 2. The nominal decrease (29) is likewise approximated by

$$s_{t+1}\|w_{t+1}\|^2 + \alpha_t E_t + (1 - \alpha_t)\hat{E}_t\,. \tag{32}$$

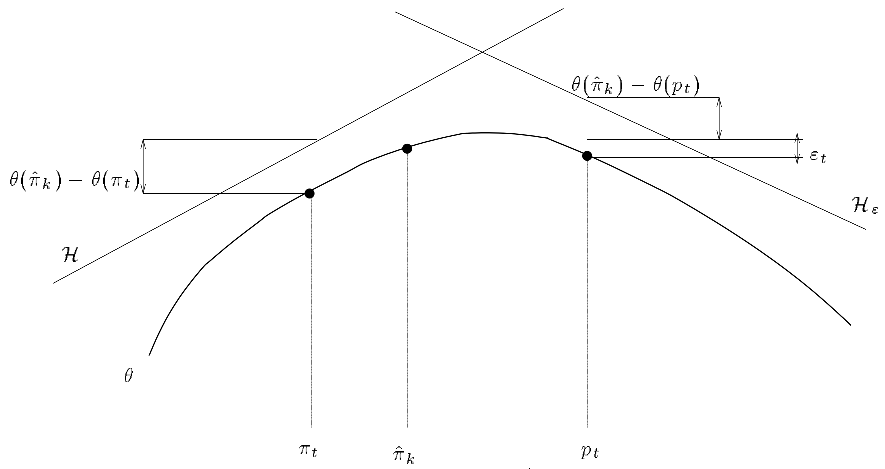


**Fig. 2.** Model $\hat{\Theta}_t$.

**Proposition 1.** *The following relation holds*

$$\alpha_t E_t + (1 - \alpha_t)\hat{E}_t = \langle w_{t+1}, \hat{\pi}_k - p_{t+1}\rangle + \varepsilon_{t+1}\,.$$

*Proof.* Let $\mathcal{E} := \alpha_t E_t + (1 - \alpha_t)\hat{E}_t$. For simplicity, we drop subindices $t$ and use $_+$ to denote subindices $t+1$. Using (30) and (31), write the lefthand side as follows

$$\begin{aligned}
\mathcal{E} &= \alpha\langle v, \hat{\pi}_k - \pi\rangle + (1-\alpha)\langle w, \hat{\pi}_k - p\rangle + (1-\alpha)\varepsilon \\
&= \langle w_+, \hat{\pi}_k\rangle - \alpha\langle v, \pi\rangle - (1-\alpha)\langle w, p\rangle + (1-\alpha)\varepsilon && \text{[by (15)]} \\
&= \langle w_+, \hat{\pi}_k - p_+\rangle + \langle w_+, p_+\rangle - \alpha\langle v, \pi\rangle - (1-\alpha)\langle w, p\rangle + (1-\alpha)\varepsilon && [\text{add } \pm\langle w_+, p_+\rangle] \\
&= \langle w_+, \hat{\pi}_k - p_+\rangle + \alpha\langle v, p_+ - \pi\rangle + (1-\alpha)\langle w, p_+ - p\rangle + (1-\alpha)\varepsilon && \text{[by (15)]} \\
&= \langle w_+, \hat{\pi}_k - p_+\rangle + \alpha(1-\alpha)\langle v - w, p - \pi\rangle + (1-\alpha)\varepsilon && \text{[by (18)]} \\
&= \langle w_+, \hat{\pi}_k - p_+\rangle + \varepsilon_+\,,
\end{aligned}$$

by (20). □

We now incorporate these new elements in the algorithmic pattern of VA.

## 5. A revised formulation of the volume method

In this section we describe the "minimalistic" approach, in which the unknown value of $\theta(p_t)$ is replaced by $\theta(\hat{\pi}_k)$. In order to make the sequence $\{\theta(\hat{\pi}_k)\}$ monotone, and based on (32) and Proposition 1, in (33) below we define the expected gain so that it is always positive. The detailed algorithm follows.

*5.1. The Revised Volume Algorithm*

STEP 0. Let $m_1 \in (0,1)$ be a given tolerance. Given an initial $\pi_0 \in \mathbb{R}^m$, compute $x_0 \in \operatorname{Argmin} L(x, \pi_0)$, a solution of (10) written for $\pi = \pi_0$. Let $v_0 := Ax_0 - b$. Initialize $z_1 = x_0$, $\hat{\pi}_1 = \pi_0$, and $w_1 := v_0$, as well as $p_1 = \pi_0$ and $\varepsilon_1 = 0$. Set $k = t = 1$ and $T_s = \emptyset$.

STEP 1. Having the center $\hat{\pi}_k$ and a steplength $s_t > 0$ make the move (12):

$$\pi_t = \hat{\pi}_k + s_t w_t\ .$$

Compute the ascent measure

$$\delta_t = s_t \|w_t\|^2 + |\langle w_t, \hat{\pi}_k - p_t\rangle| + \varepsilon_t\ . \tag{33}$$

STEP 2. Depending on whether

$$\theta(\pi_t) \geq \theta(\hat{\pi}_k) + m_1 \delta_t \tag{34}$$

is false or true, decide to make, respectively,

– either a *null-step*: (34) does not hold, just do nothing,

– or a *serious-step*: (34) holds. Update the stability center: $\hat{\pi}_{k+1} = \pi_t$, set $t_k := t$, $T_s = T_s \cup \{t_k\}$, and increase $k$ by 1.

STEP 3. Compute $x_t \in \operatorname{Argmin} L(x, \pi_t)$, a solution of (10) written for $\pi = \pi_t$. Let $v_t := Ax_t - b$.

STEP 4. Compute a new stepsize $s_{t+1}$. Let $\alpha_t \in [0,1]$ be the solution of (28) with $e_t$ and $\hat{e}_t$ therein replaced by $E_t$ and $\hat{E}_t$ from (30) and (31), respectively.

Compute

$$\begin{aligned} z_{t+1} &= \alpha_t x_t + (1-\alpha_t) z_t \\ w_{t+1} &= \alpha_t v_t + (1-\alpha_t) w_t \\ p_{t+1} &= \alpha_t \pi_t + (1-\alpha_t) p_t \\ \varepsilon_{t+1} &= \alpha_t \sigma_t + (1-\alpha_t)\varepsilon_t\ , \end{aligned}$$

where $\sigma_t$ is defined in (21). This completes the $t^{th}$ iteration: set $t = t+1$ and loop to 1. □

To get a better understanding of the sequence generated by RVA, we refer again to Figure 2. Note that now the hyperplanes $\mathcal{H}$ and $\mathcal{H}_\varepsilon$ only support the graph of $\theta$ at $\pi_t$ within $\theta(\hat{\pi}_k)$, and at $p_t$ within $\theta(\hat{\pi}_k) + \varepsilon_t$, respectively. We know that $\theta(\hat{\pi}_k) \geq \theta(\pi_t)$ for all $t \leq t_k$, so, when compared to Figure 1, $\mathcal{H}$ will always be shifted to the outside of *graph* $\theta$. For the hyperplane $\mathcal{H}_\varepsilon$ this may not always be the case: if $\theta(\hat{\pi}_k) < \theta(p_t) + \varepsilon_t$, then $\mathcal{H}_\varepsilon$ will be shifted down, towards *graph* $\theta$. In other words, with this model the inequality $\hat{\Theta}_t(\pi) \geq \theta(\pi)$ may not hold for all $\pi \in \mathbb{R}^m$. As a result, we might be *cutting off* a region containing a maximum of $\theta$. This nasty feature explains why we have not been able to get a satisfactory convergence result when RVA generates only a finite number of serious steps, see Lemma 4 below. In order to get a full convergence result, we need to make additional changes in RVA. We will come back to this point in Section 6.2.

*5.2. Convergence properties*

Even though RVA's model $\hat{\Theta}_t$ may not support the whole graph of $\theta$, this revised version of VA represents an improvement in terms of convergence properties. We start with a result on boundedness.

**Lemma 3.** *Let* (1) *be such that* $\Psi$ *from* (3) *is bounded. Then the following sequences generated by RVA are bounded:*

$$\{x_t\}, \{v_t\}, \{z_t\}, \{w_t\}.$$

*Furthermore, if* $\{\pi_t\}$ *is bounded, so is* $\{p_t\}$.

*Proof.* Since by definition $x_t \in \Psi$ for all $t$, our assumption on $\Psi$ implies that the sequence $\{x_t\}$ is bounded. Thus, so is the sequence $\{v_t = Ax_t - b\}$ defined in STEP 3 of RVA. Together with Lemma 2$(ii)$, this implies that the sequences $\{z_t\}$ and $\{w_t\}$ are bounded. Finally, from (19), the same result applies for $\{p_t\}$, provided $\{\pi_t\}$ is bounded. □

As usually done in bundle methods, for our analysis we first suppose the cardinality of $T_s$ is infinite. To prove convergence in this situation, we give general conditions on the stepsize $s_t$, updated at STEP 4 of RVA.

**Theorem 2.** *Suppose that RVA generates infinitely many serious steps, i.e,* $k \to \infty$. *The following holds:*

*(i) Either* $\theta(\hat\pi_k) \to +\infty$ *or* $\lim_{t \in T_s} \delta_t = 0$.

*(ii) As a result,* $\lim_{t\in T_s} \varepsilon_t = 0$.

*(iii) Suppose, in addition, that*

$$\sum_{t \in T_s} s_t = +\infty. \tag{35}$$

*Then* $\lim_{t\in T_s} w_t = 0$.

*(iv) Suppose, in addition to* (35), *that*

$$\mathcal{P}^\star, \textit{the solution set of } (5), \textit{ is non empty,} \tag{36}$$

*and*

$$\textit{the sequence } \{s_t\} \textit{ is bounded.} \tag{37}$$

*Then the sequence* $\{\hat\pi_k\}$ *is bounded.*

*(v) Suppose, in addition to* (35), (36) *and* (37), *that*

$$\textit{the sequence } \{w_t\} \textit{ is bounded.} \tag{38}$$

*Then the whole sequence* $\{\pi_t\}$ *is bounded.*

*Proof.* [$(i)$] Suppose $\{\theta(\hat\pi_k)\} \not\to +\infty$. Then summing (34) over $k$ implies that

$$m_1 \sum_{k=1}^{\infty} \delta_{t_k} \le \sum_{k=1}^{\infty} \Big(\theta(\hat\pi_{k+1}) - \theta(\hat\pi_k)\Big) = \lim_k \theta(\hat\pi_k) - \theta(\hat\pi_1) < +\infty.$$

Since $\sum_{t\in T_s} \delta_t = \sum_{k=1}^{\infty} \delta_{t_k}$ is finite, the result follows.
[$(ii)$] By (33), $0 \le \varepsilon_t \le \delta_t$, so this item is straightforward from $(i)$.
[$(iii)$] Use again (33) to see that $s_t\|w_t\|^2 \le \delta_t$ for all $t$. Thus, using $(i)$, the series $\sum_{t\in T_s} s_t\|w_t\|^2$ is finite and, by (35), this means that $\liminf_{t\in T_s} \|w_t\|^2 = 0$.

[$(iv)$] Take $p^\star \in \mathcal{P}^\star \neq \emptyset$, by (36). Since $w_{t_k} \in \partial_{\varepsilon_{t_k}}\theta(p_{t_k})$, the $\varepsilon$-supergradient inequality (8) written with $(\pi', v, p, \varepsilon) = (p^\star, w_{t_k}, p_{t_k}, \varepsilon_{t_k})$ yields

$$\theta(p^\star) \leq \theta(p_{t_k}) + \langle w_{t_k}, p^\star - p_{t_k}\rangle + \varepsilon_{t_k} \Rightarrow \langle w_{t_k}, p_{t_k} - p^\star\rangle \leq \theta(p_{t_k}) - \theta(p^\star) + \varepsilon_{t_k} \leq \varepsilon_{t_k} . \tag{39}$$

Letting $\Pi := \|\hat\pi_{k+1} - p^\star\|^2$, write the following algebraic steps

$$\begin{aligned}
\Pi &= \|\hat\pi_k - p^\star\|^2 + \|\hat\pi_{k+1} - \hat\pi_k\|^2 + 2\langle \hat\pi_{k+1} - \hat\pi_k, \hat\pi_k - p^\star\rangle && [\text{add } \pm\hat\pi_k]\\
&= \|\hat\pi_k - p^\star\|^2 + s_{t_k}^2\|w_{t_k}\|^2 + 2s_{t_k}\langle w_{t_k}, \hat\pi_k - p^\star\rangle && [\text{use (12)}]\\
&= \|\hat\pi_k - p^\star\|^2 + s_{t_k}\left(s_{t_k}\|w_{t_k}\|^2 + 2\langle w_{t_k}, \hat\pi_k - p_{t_k}\rangle + 2\langle w_{t_k}, p_{t_k} - p^\star\rangle\right) && [\text{add } \pm p_{t_k}]\\
&\leq \|\hat\pi_k - p^\star\|^2 + s_{t_k}\left(s_{t_k}\|w_{t_k}\|^2 + 2\langle w_{t_k}, \hat\pi_k - p_{t_k}\rangle + 2\varepsilon_{t_k}\right) && [\text{use (39)}]\\
&\leq \|\hat\pi_k - p^\star\|^2 + 2s_{t_k}\delta_{t_k} . && [\text{use (33)}]
\end{aligned}$$

Together with (37), there exists some $s_{\max} > 0$ such that

$$\|\hat\pi_{k+1} - p^\star\|^2 \leq \|\hat\pi_k - p^\star\|^2 + 2s_{\max}\delta_{t_k} .$$

Since the series $\sum_{t_k}\delta_{t_k}$ converges, the result follows.
[$(v)$] In view of $(iv)$, we only need to check boundedness of the subsequence of null steps. Between two serious steps, say $k$ and $k+1$, there is a finite number of null steps indexed by $t$, with $t_{k-1} < t < t_k$. For any such $t$, from (12), $\pi_t = \hat\pi_k + s_t w_t$. Hence, $\|\pi_t - \hat\pi_k\| = s_t\|w_t\| \leq s_{max}V_{max}$, where $s_{max}$ and $V_{max}$ are the bounds given, respectively, by (37) and (38). Then, for each $k$, the subset of null steps $\{\pi_t\}_{t_{k-1}<t<t_k}$ is in a ball $B(\hat\pi_k, s_{max}V_{max})$. Since the radius is uniform in $k$, using $(iv)$ the proof is complete. ☐

The following corollary summarizes the convergence result obtained so far.

**Corollary 1.** *Let* (1) *be such that $\Psi$ from* (3) *is bounded. Assume that $\mathcal{P}^\star$, the solution set of* (5), *is nonempty and $\mathrm{dom}\,\theta = \mathbb{R}^m$. Suppose that RVA generates infinitely many serious steps. If the stepsizes $s_t$ are chosen so that* (35) *and* (37) *hold, then the sequence $\{p_t\}_{t\in T_s}$ is maximizing.*

*Proof.* By Lemma 3, (38) holds, so Theorem 2$(v)$ applies: the sequence $\{\pi_t\}$ is bounded. Again by Lemma 3, this implies that the sequence $\{p_t\}$ is bounded. Therefore, the (bounded) subsequence $\{p_t\}_{t\in T_s}$ has a limit point $p^\star$. Using items $(iii)$ and $(ii)$ of Theorem 2, we obtain from (9) that $0 \in \partial\theta(p^\star)$. ☐

Note that the original stepsize in the Volume Algorithm (cf. (16)) is consistent with conditions (35) and (37), typical in subgradient methods.

If the cardinality of $T_s$ is finite, there is a last serious step $\hat\pi := \hat\pi_{k_{\text{last}}} = \hat\pi_{t_{\text{last}}}$, followed by an infinite number of null steps. In this case, we only have a partial answer, depending essentially on a rather strong assumption, namely (40) below, where we require two (bounded) sequences to have *unique* accumulation points.

**Lemma 4.** *Assume RVA generates finitely many serious steps. The following holds:*

*(i) If* (37) *and* (38) *hold, the sequence* $\{\pi_t\}$ *is bounded.*

*(ii) Suppose, in addition, that*

$$\text{the sequences } \{\pi_t\} \text{ and } \{v_t\} \text{ converge to } \pi^\star \text{ and } v^\star, \text{ respectively.} \tag{40}$$

*Take* $m_1$ *in RVA such that* $m_1 \in (0, \frac{1}{2})$ *and suppose that*

$$\exists s_{min} > 0 \text{ such that the sequence of stepsizes } \{s_t\}_{t \notin T_s} \text{ converges to } s_{min}. \tag{41}$$

*Then both* $\pi^\star$ *and* $\hat{\pi}$ *solve* (5).

*Proof.* Since for all $t > t_{\mathtt{last}}$, $\hat{\pi}$ remains fixed, item $(i)$ is straightforward from our assumptions.

To prove $(ii)$, first note that by (19) and (40), Silverman-Toeplitz's Theorem applies (see for instance [11, Chapter II Theorem 2]):

$$\lim p_t = \lim \pi_t = \pi^\star \quad \text{and} \quad \lim w_t = \lim v_t = v^\star .$$

Moreover, the closedness of $\partial\theta$ implies that $v^\star \in \partial\theta(\pi^\star)$. In addition, (40), together with (12), (38), and (41), implies that

$$\pi^\star = \lim \pi_t = \lim \hat{\pi} + s_t w_t = \hat{\pi} + s_{min} v^\star . \tag{42}$$

Consider now the sequence $\{\varepsilon_t\}$ and recall (21). Since with our assumptions $\sigma_t \to 0$, by Toeplitz's Theorem,

$$\lim \varepsilon_t = 0 .$$

This result, combined with (42), yields that, in the limit, the expected gain from (33) satisfies

$$\lim \delta_t = \lim s_t \|w_t\|^2 + |\langle w_t, \hat{\pi} - p_t \rangle| + \varepsilon_t = 2 s_{min} \|v^\star\|^2 .$$

Because for all $t > t_{\mathtt{last}}$ only null steps are done, (34) never holds: $\theta(\pi_t) < \theta(\hat{\pi}) + m_1 \delta_t$. Passing to the limit,

$$\theta(\pi^\star) \leq \theta(\hat{\pi}) + 2 m_1 s_{min} \|v^\star\|^2 .$$

Since $v^\star \in \partial\theta(\pi^\star)$, the supergradient inequality (7) written at $\pi' = \hat{\pi}$ and (42) imply that

$$\theta(\hat{\pi}) \leq \theta(\pi^\star) + \langle v^\star, \hat{\pi} - \pi^\star \rangle = \theta(\pi^\star) - s_{min} \|v^\star\|^2 .$$

Altogether,

$$\theta(\pi^\star) \leq \theta(\hat{\pi}) + 2 m_1 s_{min} \|v^\star\|^2 \leq \theta(\pi^\star) + (2 m_1 - 1) s_{min} \|v^\star\|^2 \tag{43}$$

Thus, $(1 - 2m_1) s_{min} \|v^\star\|^2 \leq 0$. Since, by assumption, $m_1 < 1/2$, the last inequality only holds if $v^\star = 0$, i.e., if $\pi^\star$ solves (5). Finally, $\hat{\pi}$ is also a maximizer, because from (43) we deduce that $\theta(\pi^\star) = \theta(\hat{\pi})$, □

We point out that, in view of the strong assumption (40), the convergence analysis of RVA cannot be considered complete. To give a full convergence analysis, we are bound to further modify VA. We consider the resulting modification, as well as how to recover primal solutions in the next section.

## 6. Further properties of RVA

The primal sequence from (14) was not yet considered in our analysis. To relate this sequence to a solution of (1), we introduce a stopping test in RVA.

### *6.1. RVA with stopping test*

Consider the following simple modifications in the revised volume algorithm:

STEP 0. Same than for RVA. A stopping-tolerance $\delta_{\min} > 0$ is also given.

STEP 1. Same than for RVA. After computing $\delta_t = s_t\|w_t\|^2 + |\langle w_t, \hat\pi_k - p_t\rangle| + \varepsilon_t$ in (33), make the stopping test:

$$\text{If } \delta_t \le \delta_{\min}, \text{ STOP.} \tag{44}$$

STEP 2. Same than for RVA.

STEP 3. Same than for RVA.

STEP 4. Same than for RVA. □

It is possible to replace the stopping test in STEP 1 by the pair of conditions:

$$\|w_t\|^2 \le \delta_w^2 \quad \text{ and } \quad |\langle w_t, \hat\pi_k - p_t\rangle| + \varepsilon_t \le \delta_\varepsilon \tag{45}$$

for $\delta_w$ , $\delta_\varepsilon$ two tolerances given at STEP 0. Moreover, if $s_t\delta_w^2 + \delta_\varepsilon \le \delta_{\min}$ , we see that (45) implies (44). A potential advantage of (45) is that it does not depend on $s_t$, which may become unduly small as $t$ increases.

When $\delta_{\min}$ is set to 0, the algorithm loops forever and we are in the framework of Section 5.2. When $\delta_{\min}$ (or $\delta_w$ , $\delta_\varepsilon$) is positive, if (40) holds, the stopping test is activated and there is a last index $t_{last}$. Our next result establishes an a-posteriori error bound relating the last generated primal approximation in (14) to a primal solution, i.e., a solution of (1).

**Proposition 2.** *Consider RVA with stopping test* (45)*. Suppose there is an index $t_{last}$ such that* (44) *occurs and let $z_{\mathtt{last}}$ be the corresponding $z_t$ generated last. If rank$A = m$ in* (1)*, then there exist $L \in (0, +\infty)$ depending only on the data of* (1) *such that, for any $x^\star$ solving* (1)*,*

$$\|x^\star - z_{\mathtt{last}}\| \le L\delta_w \,.$$

*Proof.* Since $x^\star$ solves (1), it holds that $Ax^\star = b$ and, thus, $A(z_{\mathtt{last}} - x^\star) = Az_{\mathtt{last}} - b = w_{\mathtt{last}}$, by Lemma 1. As a result,

$$\|x^\star - z_{\mathtt{last}}\| = \|(A^\top A)^{-1}A^\top A(x^\star - z_{\mathtt{last}})\| = \|(A^\top A)^{-1}A^\top w_{\mathtt{last}}\| \le L\|w_{\mathtt{last}}\| \,.$$

Together with (45), the result follows. □

The proposition above just shows that $z_{\mathtt{last}}$ is at distance smaller than $L\delta_w$ from the feasible set $\{x \in \Psi : Ax = b\}$, containing the solution set of (1).

*6.2. A bundle method derived from RVA*

We now show how to transform RVA in a bundle method. The key is to organize the calculations so that the model $\hat\theta_t$ used in (26) never cuts off a section of $graph\,\theta$, i.e., such that $\hat\theta_t(\pi) \geq \theta(\pi)$ for all $\pi \in \mathbb{R}^m$. If this is the case, we have

$$\begin{aligned}\theta(\pi) \leq \hat\theta_t(\pi) &\leq \hat\theta_t(\pi_{t+1}) + \langle w_{t+1}, \pi - \pi_{t+1}\rangle \qquad (\star)\\ &= \theta(\hat\pi_k) + \langle w_{t+1}, \pi - \hat\pi_k\rangle + \Big(\hat\theta_t(\pi_{t+1}) - \theta(\hat\pi_k) + \langle w_{t+1}, \hat\pi_k - \pi_{t+1}\rangle\Big).\end{aligned}$$

for all $\pi \in \mathbb{R}^m$. Since by (27), $\langle w_{t+1}, \hat\pi_k - \pi_{t+1}\rangle = s_{t+1}\|w_{t+1}\|^2$,

$$\theta(\pi) \leq \theta(\hat\pi_k) + \langle w_{t+1}, \pi - \hat\pi_k\rangle + \mathcal{E}_{t+1}$$

where we defined $\mathcal{E}_{t+1} := \hat\theta_t(\pi_{t+1}) - \theta(\hat\pi_k) - s_{t+1}\|v_{t+1}\|^2$ .Writing $(\star)$ at $\pi = \hat\pi_k$ yields that $\mathcal{E}_{t+1} \geq 0$ and, thus, $w_{t+1} \in \partial_{\mathcal{E}_{t+1}}\theta(\hat\pi_k)$. Therefore, the update (12) can now be interpreted as an approximate supergradient move, no extragradient point $p_t$ is involved.

Accordingly, the model used in the calculations will be

$$\hat\theta_t(\pi) := \theta(\hat\pi_k) + \min\{e_t + \langle v_t, \pi - \hat\pi_k\rangle\,, \mathcal{E}_t + \langle w_t, \pi - \hat\pi_k\rangle\}\,,$$

with $\mathcal{E}_1 = 0$. Subsequent $\mathcal{E}_{t+1}$ can be computed using the definition above, or using $\alpha_t$, as described next. The primal optimal value giving $\alpha_t$ in (26) is

$$\hat\theta_t(\pi_{t+1}) - \frac{1}{2s_{t+1}}\|\pi_{t+1} - \hat\pi_k\|^2 = \hat\theta_t(\pi_{t+1}) - \frac{s_{t+1}}{2}\|w_{t+1}\|^2\,,$$

by (27). The dual value from (28) is

$$\frac{s_{t+1}}{2}\|\alpha_t v_t + (1-\alpha_t)w_t\|^2 + \alpha_t e_t + (1-\alpha_t)\mathcal{E}_t = \frac{s_{t+1}}{2}\|w_{t+1}\|^2 + \alpha_t e_t + (1-\alpha_t)\mathcal{E}_t\,,$$

by (15). Problem (28) is strongly convex, so both optimal values are equal and

$$\mathcal{E}_{t+1} = \hat\theta_t(\pi_{t+1}) - \hat\theta_t(\pi_{t+1}) - s_{t+1}\|w_{t+1}\|^2 = \alpha_t e_t + (1-\alpha_t)\mathcal{E}_t\,.$$

The corresponding modifications in the revised volume algorithm of Section 5.1 yield the following scheme, that we call BVA:

STEP 0. Same than for RVA. Replace $\varepsilon_1 = 0$ by $\mathcal{E}_1 = 0$.

STEP 1. Same than for RVA, replacing (33) by

$$\delta_t = s_t\|w_t\|^2 + \mathcal{E}_t\,. \qquad (33)'$$

STEP 2. Same than for RVA.

STEP 3. Same than for RVA.

STEP 4. Compute a new stepsize $s_{t+1}$. Let $\alpha_t \in [0,1]$ be the solution of (28) with $\varepsilon_t$ as defined in (22) and $\hat e_t$ replaced by $\mathcal{E}_t$. Compute

$$\begin{aligned} w_{t+1} &= \alpha_t v_t + (1-\alpha_t)w_t\\ \mathcal{E}_{t+1} &= \alpha_t e_t + (1-\alpha_t)\mathcal{E}_t\,.\end{aligned}$$

This completes the $t^{th}$ iteration: set $t = t+1$ and loop to 1. □

This algorithm falls within the class of penalized bundle methods, implemented with maximum bundle compression at each iteration. More specifically, BVA is Algorithm 3.14 in [7, Ch. XV], with maximum bundle size equal to 2. For a proof of convergence, we refer to Theorems 3.2.2 and 3.2.4 therein. Barring the objectionable condition (40), it is interesting to compare the assumptions on the stepsizes $s_t$ required by these theorems with our own assumptions for RVA:

| ALGORITHM | $\infty$ SERIOUS STEPS | $\infty$ NULL STEPS |
|---|---|---|
| RVA | (35) and (37) | (37) and (41) |
| BVA | (35) and $\{s_t\}_{t\in T_s}$ bounded | $\{s_t\}_{t\notin T_s}$ non increasing and $\sum_{t\notin T_s} \frac{s_{t+1}^2}{s_t} = \infty$ |

Our condition (37) is implied by $\{s_t\}_{t\in T_s}$ bounded together with $\{s_t\}_{t\notin T_s}$ non increasing. On the other hand, we do not require this last monotony assumption on the subsequence $\{s_t\}_{t\notin T_s}$. Finally, when compared to (41), the two conditions for proving convergence for infinite null steps in BVA imply that the decreasing sequence $\{s_t\}_{t\notin T_s}$ is bounded away from zero.

We finish by mentioning that a result along the lines of Proposition 2 can also be proved for the sequence $\{z_t\}$ computed in STEP 4 of BVA. In fact, for such a result to hold, it is only required from the algorithm to have a stopping test measuring $\|w_t\|$ like in (44), or (45), with $w_t \in \partial\theta(z_t)$.

## 7. Computational experience

We now compare VA, RVA and BVA on a set of Rectilinear Steiner problems.

### *7.1. Formulation of the problem*

Given an undirected weighted graph $G = (V, E)$, a set of terminal vertices $T \subseteq V$, with $|T| \geq 3$ and non-negative edge weights $c_e$ for all $e \in E$, the *Steiner Problem in Graphs* consists in finding a connected subgraph $S$ of $G$ (called the *Steiner Tree*) that includes all terminal vertices at minimum edge cost, i.e., $\min \sum_{e\in S} c_e$. This problem is known to be $\mathcal{NP}$-hard, specially for grid graphs, see [10], [6]. In this section we report computational results on *Rectilinear Steiner Problems*, i.e., instances in which the graph $G$ is a grid, possibly with holes. Rectilinear instances are known to be the hardest ones, and the existence of holes makes the problem even harder.

Motivated by the increasing demand in the *VLSI design* of electronic circuits, the solution of Steiner problems has received considerable attention in the last years. Among the proposed solution methods are cutting-planes algorithms, heuristic procedures, approximation algorithms, Lagrangian relaxations, and polyhedral approaches. Related surveys are [20], [15], [8], [9] and, more recently, [12] and [3].

We use VA, RVA and BVA to solve a linear relaxation of the *nonsimultaneous single-commodity flow integer formulation* of the Steiner problem from [5], [22]. In order to obtain this formulation, we first express the original (undirected weighted graph) problem as a directed weighted graph problem. We proceed as follows:

- For each edge $e = [i, j] \in E$ we create arcs $(i, j)$ and $(j, i) \in E_d$ with $c_{ij} = c_{ji} = c_e$, yielding the directed graph $G_d = (V, E_d)$;
- We choose one vertex $s \in T$ to be the *source* offering $T_0 := |T \setminus \{s\}|$ commodities, one commodity for each of the remaining $|T_0|$ terminal vertices;
- For $k = 1, \dots, T_0$ and $(i, j) \in E_d$, let $f^k_{ij}$ be the amount of commodity $k$ on the arc $(i, j)$, and let $x_{ij}$ be a binary variable indicating whether the arc $(i, j)$ is in the Steiner Tree ($x_{ij} = 1$) or not ($x_{ij} = 0$);
- Consider further that $I(i)$ is the set of all (input) vertices $j \in V$ such that $(j, i) \in E_d$, and $O(i)$ is the set of all (output) vertices $j \in V$ such that $(i, j) \in E_d$.

Altogether, defining $\mathcal{A} := \{(x, f) = (x_{ij}, f^k_{ij}) : (i, j) \in E_d\,, k = 1, \dots, T_0\}$ and $c := (c_{ij})_{(i,j)\in E_d}$, we obtain

$$
\left\{
\begin{array}{ll}
\min\limits_{(x,f)\in\mathcal{A}} \ \langle c, x\rangle & \\
\sum\limits_{j\in O(s)} f^k_{sj} - \sum\limits_{j\in I(s)} f^k_{js} = 1\,, \quad k = 1, \dots, T_0 & (\mathrm{a}_1)\\
\sum\limits_{j\in O(k)} f^k_{kj} - \sum\limits_{j\in I(k)} f^k_{jk} = -1\,, \quad k = 1, \dots, T_0 & (\mathrm{a}_2)\\
\sum\limits_{j\in O(i)} f^k_{ij} - \sum\limits_{j\in I(i)} f^k_{ji} = 0\,, \quad i \in V \setminus \{s, k\}, \quad k = 1, \dots, T_0 & (\mathrm{a}_3)\\
x_{ij} \in \{0, 1\}\,, \quad (i, j) \in E_d\,. & (\mathtt{int})\\
0 \le f^k_{ij} \le x_{ij}\,, \quad (i, j) \in E_d\,, \quad k = 1, \dots, T_0 & (\Psi).
\end{array}
\right.
\tag{46}
$$

Constraints (46)($\mathrm{a}_1$), ($\mathrm{a}_2$), and ($\mathrm{a}_3$) represent the flow conservation equations for, respectively, the terminal vertex chosen to be the source, the remaining terminal vertices, and the non-terminal vertices. The constraint set (46)($\Psi$) allows a nonzero flow $f^k_{ij}$ of any commodity $k$ through an arc $(i, j)$ only if this arc is included in the Steiner Tree. Finally, the objective function is defined as the total sum of the arcs included in the Steiner Tree, i.e., arcs $(i, j)$ such that $x_{ij} = 1$. In particular, since the costs $c_{ij}$ are integer, this means that the objective function $\langle c, x\rangle$ only takes integer values.

*7.2. Dual problem*

We now derive the corresponding dual problem (5). To obtain a continuous linear program as in (1) we consider, instead of (46), its linear relaxation, i.e., a problem with (46)(`int`) replaced by

$$x_{ij} \in [0,1]\,, \quad (i,j) \in E_d\,. \qquad (46)(\texttt{lin})$$

We have relaxed constraints (46)($a_1$)-($a_3$), playing the role of constraints (1)(a). Accordingly, the number of dual and primal variables are, respectively

$$m := (1+1+|V|-2)T_0 = |V|(|T|-1) \text{ and } n := |\mathcal{A}| = 2|E_d| + 2|E_d|T_0 = 2|E_d||T|\,.$$

For convenience, we consider for each dual variable $\pi \in \Re^m$ subvectors $\pi^k \in \Re^{|V|}$, with $k = 1, \ldots, T_0$. With this notation the dualization of, for example, (46)($a_1$) with $k$ fixed, gives a scalar term $\pi_s^k \left(\sum_{j \in O(s)} f_{sj}^k - \sum_{j \in I(s)} f_{js}^k - 1\right)$ in the Lagrangian function (2). The righthand side vector $b \in \Re^m$ in (1) is defined by $b_s^k := 1$, $b_k^k := -1$ and $b_i^k = 0$ for all $i \neq s\,, k$ and for all $k = 1, \ldots, T_0$. The $m \times n$ matrix $A$ from (1) can be defined likewise.

The remaining constraints, (46)(`lin`) and (46)($\Psi$), form the feasible polyhedron $\Psi$ from (3). Altogether, the dual function from (10) has the expression

$$\theta(\pi) = \begin{cases} \displaystyle\min_{(x,f)\in\mathcal{A}} \ \langle c, x\rangle + \sum_{k=1}^{T_0} \sum_{(i,j)\in E_d} (\pi_i^k - \pi_j^k) f_{ij}^k - \langle b, \pi\rangle \\ \left.\begin{array}{ll} x_{ij} \in [0,1]\,, & (i,j) \in E_d \\ 0 \leq f_{ij}^k \leq x_{ij}\,, & (i,j) \in E_d\,, k = 1, \ldots, T_0\,. \end{array}\right] =: \Psi \end{cases}$$

Each call to the oracle entails solving this subproblem by inspection on $\Psi$. Namely, letting $\ell_{ij}^k := \pi_i^k - \pi_j^k$, a solution $(x(\pi), f(\pi))$ satisfies

$$\text{If } \sum_{k:\,\ell_{ij}^k<0} |\ell_{ij}^k| \ > c_{ij}\,, \ \text{ then } \begin{cases} x_{ij}(\pi) := 1\,, \\ f_{ij}^k(\pi) := 1 \quad \text{for every } k \text{ such that } \ell_{ij}^k < 0\,, \\ f_{ij}^k(\pi) := 0 \quad \text{for every } k \text{ such that } \ell_{ij}^k \geq 0\,. \end{cases}$$

$$\text{If } \sum_{k:\,\ell_{ij}^k<0} |\ell_{ij}^k| \ \leq c_{ij}\,, \ \text{ then } \begin{cases} x_{ij}(\pi) := 0\,, \\ f_{ij}^k(\pi) := 0 \quad \text{for every } k\,. \end{cases}$$

Finally, by (11), the supergradient $v = A\Big(x(\pi), f(\pi)\Big) - b$ is readily available.

*7.3. Description of the solution method*

We now give the general algorithmic scheme we use for all the three dual methods VA, RVA, BVA. We start with a description on how to compute upper bounds.

*Upper bounds.* Let $z_t = (\hat{x}_t, \hat{f}_t)$ be the primal approximation (of a solution of the linear programming relaxation of (46)) computed in VA, RVA and BVA using (14). We use this primal information to derive heuristics yielding an upper bound (i.e., an integer feasible point) for problem (46).

In what follows we denote by $\hat{x}$ the vector $\hat{x}_t$ and for every edge $(i,j) \in G = (V,E)$, we let $\hat{y}_{ij} := \hat{x}_{ij} + \hat{x}_{ji}$.

*1.- Minimum spanning tree with volumetric weights (MSTV).* This heuristic is defined by the following steps:

- Find a minimum spanning tree $MSTV$ in the graph $G$ with arc weights equal to $c_{ij}$ if $t = 0$ and $-\hat{y}_{ij}$ for all $t > 0$.
- Prune all non-terminal leaves of $MSTV$.
- Find a minimum spanning tree $MST$ in the subgraph induced by the vertices remaining in $MSTV$ after the pruning, considering the original costs $c_{ij}$ as weights.
- Prune all non-terminal leaves of MST.

*2.-Minimum spanning tree in a modified graph (MSTM).* This heuristic is applied in a subgraph of $G$. For a given $\beta \in [0,1]$, let $G_\beta = (V_\beta, E_\beta)$ be the subgraph induced by the set of vertices $V_\beta$ consisting of all terminal vertices and the nonterminal vertices $i$ satisfying $\sum_j \hat{y}_{ij} \geq \beta$ . The steps are:

- Find the largest value of $\beta \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$ such that $G_\beta$ is connected.
- Find a minimum spanning tree $MSTM$ in $G_\beta$ with arc weights equal to $c_{ij}$ if $t = 0$ and $(1 - \hat{y}_{ij})c_{ij}$ for all $t > 0$.
- Prune all non-terminal leaves of $MSTM$.
- Find a minimum spanning tree $MST$ in the graph induced by the vertices remaining in $MSTM$ after the pruning, considering the original costs $c_{ij}$ as weights.
- Prune all non-terminal leaves of $MST$.

*3.- Takahashi & Matsuyama heuristic with volumetric weights (T&MV).* Given a graph $G = (V,E)$ with nonnegative arc costs $c_{ij}$, for each $(i,j) \in E$, the *Takahashi & Matsuyama* heuristic is defined in [18] by:

1. Choose an initial terminal vertex $v_i$ and set $k = 1$.
2. Connect $v_i$ with the terminal vertex $v_j$, $j \neq i$, that is closest to $v_i$ using the shortest path. Let $T_1$ be the subtree obtained.
3. Connect $T_k$ with the terminal vertex $v_l$, $l \notin T_k$, that is closest to $T_k$ using the shortest path. Let $T_{k+1}$ be the subtree obtained;
4. Stop if all terminals are connected; otherwise set $k \leftarrow k + 1$ and loop to 3.

Our third heuristic performs the following steps:

- Given a starting vertex $v_i$, run the *Takahashi & Matsuyama* heuristic for the digraph $D = (V, E_d)$ with arc weights equal to $c'_{ij} := c'_{ji} := c_{ij}$ if $t = 0$ and $c'_{ij} := c'_{ji} := (1 - \hat{y}_{ij})c_{ij}$ for all $t > 0$.

– Find a minimum spanning tree in the subgraph induced by the vertices included in the tree obtained in the previous step, considering the original costs $c_{ij}$ as weights.
– Prune all non-terminal leaves.

The heuristics above are used to compute, at certain iterations of the general algorithm, three primal feasible points $\hat{x}$. The upper bound UB used in (16) is the minimum of the three corresponding objective values $\langle c, \hat{x}\rangle$ in (1).

*Stopping tests.* Since (46) has integer cost function depending only on the 0–1 variables $x_{ij}$, its optimal value is integer. As a result, at every iteration, say $t$, UB is an integer value, bigger, by weak duality, than the dual value LB$:=\theta(\pi_t)$, possibly non integer. This observation yields the stopping test

$$\text{If} \quad \text{UB} - \text{LB} < 1 \qquad \text{STOP}, \tag{IDG}$$

that we call IDG, by integer duality gap. When a method stops with this test, the solution found is optimal.

Because VA does not have a stopping criterion, we check in this case primal and dual feasibility:

$$\text{If} \quad |\langle c, \hat{x}_t\rangle - \text{LB}| \leq \text{TOL}_p\text{LB} \quad \text{and} \quad ||w_t|| \leq \text{TOL}_w \quad \text{STOP}, \tag{PDF}$$

for given initial tolerances, $\text{TOL}_{p,w}$.

As a final exit, we also set a maximum of iterations and CPU time:

$$\text{If} \quad t > \text{MAXITER} \quad \text{or} \quad \text{CPU}_{\text{time}} > \text{MAXTIME} \qquad \text{STOP}. \tag{!!}$$

*General Algorithm.*

START. Choose the stopping tolerances and the initial parameters of the dual method to run (VA, RVA, BVA). Set $t = 0$.
LOWER BOUND. Make one iteration of the dual method. Update LB. The primal approximation $z_t$ is available.
UPPER BOUND. If the iteration gives a serious step, or if too many null steps have been done, run the three heuristics, starting from $z_t$. Update UB and define a new stepsize $s_{t+1}$.
STOPPING TEST. Check (IDG). For VA, check (PDF), for RVA and BVA, check, for instance, (45). Check (!!).
LOOP. This completes the $t^{th}$ iteration: set $t = t+1$ and loop to LOWER BOUND.
□

### 7.4. Numerical results

Now we report on the computational experiences. Our code is implemented in C++ and all runs have been performed on a Pentium 133MHz.

| *Name* | $\|V\|$ | $\|E\|$ | $\|T\|$ | $n = \|PrimalPb\|$ | $m = \|DualPb\|$ |
|---|---|---|---|---|---|
| dmxa1721 | 4 | 5 | 3 | 20 | 8 |
| dmxa1109 | 9 | 13 | 5 | 104 | 36 |
| dmxa0903 | 53 | 90 | 7 | 1080 | 318 |
| dmxa0848 | 34 | 54 | 11 | 1080 | 340 |
| dmxa1200 | 29 | 42 | 13 | 1008 | 348 |
| dmxa0368 | 47 | 76 | 9 | 1216 | 376 |
| dmxa1801 | 310 | 553 | 17 | 17696 | 4960 |
| taq0631 | 8 | 11 | 4 | 66 | 24 |
| taq0023 | 37 | 63 | 7 | 756 | 222 |
| taq0739 | 64 | 105 | 12 | 2310 | 704 |
| taq0431 | 108 | 188 | 10 | 3384 | 972 |
| taq0741 | 84 | 140 | 14 | 3640 | 1092 |
| taq0751 | 104 | 178 | 14 | 4628 | 1352 |
| taq0365 | 984 | 1771 | 21 | 70840 | 19680 |
| alue2087 | 40 | 65 | 13 | 1560 | 480 |
| alue5067 | 300 | 504 | 38 | 37296 | 11100 |
| gap1904 | 7 | 9 | 4 | 54 | 21 |
| gap2740 | 13 | 19 | 5 | 152 | 52 |
| gap3100 | 16 | 25 | 8 | 350 | 112 |
| gap3036 | 28 | 42 | 9 | 672 | 224 |
| gap2007 | 41 | 70 | 9 | 1120 | 328 |
| msm1931 | 4 | 5 | 3 | 20 | 8 |
| msm2705 | 4 | 5 | 3 | 20 | 8 |
| msm1844 | 6 | 8 | 4 | 48 | 18 |
| msm0580 | 7 | 9 | 4 | 54 | 21 |
| msm0920 | 7 | 9 | 4 | 54 | 21 |
| msm1234 | 7 | 9 | 4 | 54 | 21 |
| msm2802 | 7 | 11 | 4 | 66 | 21 |
| msm2326 | 9 | 12 | 5 | 96 | 36 |
| msm2525 | 11 | 15 | 6 | 150 | 55 |
| msm1477 | 12 | 16 | 6 | 160 | 60 |
| msm1008 | 13 | 18 | 6 | 180 | 65 |
| msm4515 | 35 | 55 | 8 | 770 | 245 |
| msm2492 | 39 | 61 | 9 | 976 | 312 |
| msm2601 | 178 | 305 | 12 | 6710 | 1958 |
| msm3829 | 338 | 594 | 10 | 10692 | 3042 |
| msm2152 | 191 | 333 | 24 | 15318 | 4393 |
| msm4312 | 1299 | 2355 | 10 | 42390 | 11691 |
| msm2846 | 347 | 595 | 58 | 67830 | 19779 |
| diw0487 | 4 | 5 | 3 | 20 | 8 |
| diw0473 | 14 | 22 | 6 | 220 | 70 |
| diw0459 | 19 | 30 | 8 | 420 | 133 |
| diw0445 | 27 | 42 | 10 | 756 | 243 |
| diw0559 | 83 | 141 | 11 | 2820 | 830 |
| diw0795 | 225 | 400 | 10 | 7200 | 2025 |
| diw0778 | 187 | 337 | 15 | 9436 | 2618 |
| diw0801 | 334 | 605 | 10 | 10890 | 3006 |
| diw0234 | 760 | 1413 | 21 | 56520 | 15200 |
| diw0819 | 1394 | 2604 | 26 | 130200 | 34850 |
| diw0820 | 1816 | 3407 | 32 | 211234 | 56296 |

**Table 1.** Size of VLSI Instances

VLSI layout applications yield Steiner Tree Problems over rectangular grid graphs with many irregularly placed holes. These instances are known to be the hardest ones to be solved by current methods. For our comparisons, we use the preprocessed version [3] of the 116 VLSI problems from the library *SteinLib* available at `ftp://ftp.zib.de/pub/mp-testdata/steinlib`. After preprocessing, 42 out of the 116 instances are solved straightforwardly. From the remaining 74 instances, we chose 50 cases, classified in 6 groups, as described in Table 1.

In our General Algorithm, we use the following initial tolerances and parameters: $\mu = 0.1$ for the stepsize (16), $m_1 = 0.001$ for the serious test (34), $\text{TOL}_p = \text{TOL}_w = 0.00001$ in (PDF), and $\delta_w = \delta_\epsilon = 0.00001$ in (45). Finally, in (!!) we set MAXITER=500,000 and MAXTIME=7,200 seconds.

The complete set of results for the 50 examples are included in the Appendix, on Tables 5, 6, and 7, corresponding, respectively, to VA, RVA, and BVA. We summarize these results in Tables 2, 3, and 4. Each table reports both primal and dual average results for each one of the 6 groups of instances.

The two first columns on *Primal Results* show the percentage of the instances solved to optimality, and $PrAp$, the (average) distance between UB and the objective values $\langle c, \hat{x}_t \rangle$ computed using the primal approximations obtained by each method. Finally, a third column $Heu$ shows the heuristic(s) giving best results in average.

The columns on *Dual Results* report the total number of subproblems solved (i.e., the total number of calls to the oracle, to evaluate the dual function and to compute a supergradient), the most frequently verified stopping test, and the total average CPU time in seconds.

| | Primal Results | | | Dual Results | | |
|---|---|---|---|---|---|---|
| Group | (%)$OPT$ | (%)$PrDist$ | $Heu$ | (#)$SubPb$ | $StopT$ | $CPU_{\text{Time}}(s)$ |
| DMXA | 100.00 | 16.49 | MSTM | 6480 | (IDG) | 349.38 |
| TAQ | 85.71 | 13.38 | MSTM | 311085 | (IDG) | 4723.29 |
| ALUE | 100.00 | 18.23 | T&MV | 2340 | (IDG) | 574.23 |
| GAP | 80.00 | 15.44 | MSTM/MSTV | 503340 | (IDG) | 158.20 |
| MSM | 83.33 | 20.22 | MSTV | 558800 | (IDG) | 14961.70 |
| DIW | 81.82 | 14.05 | MSTM/MSTV | 18480 | (IDG) | 15244.00 |

**Table 2.** Summary of primal and dual results (VA)

| | Primal Results | | | Dual Results | | |
|---|---|---|---|---|---|---|
| Group | (%)$OPT$ | (%)$PrDist$ | $Heu$ | (#)$SubPb$ | $StopT$ | $CPU_{\text{Time}}(s)$ |
| DMXA | 100.00 | 15.17 | MSTM/MSTV | 6896 | (IDG) | 369.19 |
| TAQ | 100.00 | 12.33 | MSTM | 268205 | (IDG) | 9094.70 |
| ALUE | 100.00 | 14.68 | T&MV | 2770 | (IDG) | 412.71 |
| GAP | 80.00 | 15.28 | MSTV | 503330 | (IDG) | 240.56 |
| MSM | 83.33 | 19.18 | MSTV | 557269 | (IDG) | 15537.75 |
| DIW | 81.82 | 15.38 | MSTV | 23503 | (IDG) | 15572.89 |

**Table 3.** Summary of primal and dual results (RVA)

| | Primal Results | | | Dual Results | | |
|---|---|---|---|---|---|---|
| Group | (%)$OPT$ | (%)$PrDist$ | $Heu$ | (#)$SubPb$ | $StopT$ | $CPU_{\text{Time}}(s)$ |
| DMXA | 100.00 | 17.00 | MSTM/MSTV | 6297 | (IDG) | 347.68 |
| TAQ | 71.43 | 13.81 | T&MV | 531927 | (IDG) | 10403.76 |
| ALUE | 100.00 | 17.16 | T&MV/MSTM | 2437 | (IDG) | 719.18 |
| GAP | 80.00 | 15.35 | MSTV | 503176 | (IDG) | 1554.05 |
| MSM | 83.33 | 21.53 | MSTV | 601998 | (IDG) | 16951.51 |
| DIW | 81.82 | 15.12 | MSTV | 18601 | (IDG) | 15436.81 |

**Table 4.** Summary of primal and dual results (BVA)

In terms of primal results, two good measures for analyzing the quality of the algorithm are the number of instances for which optimality was found, and the quality of the primal approximation produced by the algorithm. From this point of view, we see that RVA behaves better than both VA and BVA, because the primal approximations generated by RVA are better in average for almost all groups. Furthermore, the primal information obtained with RVA resulted in optimality for all the problems of the group TAQ. More precisely, Table 6 in the Appendix shows that the stopping test (IDG) held for all the problems, but TAQ0739. Yet, for this problem, RVA gave a primal approximation value $\langle c, \hat{x}_t \rangle$ (693.32) that can be considered optimal (693).

In terms of dual results, a good measure for the algorithm quality is the number of subproblems solved. In this case, RVA also behaves better than both VA and BVA.

Finally, in terms of CPU time, VA is 10% faster than RVA, that is in turn 10% faster than BVA.

We conclude from tables 2 , 3 and 4 that RVA is better suited to solve difficult instances coming from real-world VLSI design problems. Moreover, since the good numerical performances of RVA are backgrounded by some convergence results, RVA can be considered as a good compromise between practice and theory. Altogether, when solving linear relaxations of difficult combinatorial problems, and based on our analysis and experience, we think that RVA should be preferred to both VA and BVA.

*Acknowledgements.* The research of the first author was supported by CNPq under Grant N$^{\underline{o}}$142401/96-0. The research of the second author was partially supported by CNPq, FAPERJ and PRONEX. The research of the third author was partially supported by FAPERJ under Grant N$^{\underline{o}}$E26/150.205/98 and by CNPq under Grant N$^{\underline{o}}$301800/96.

The code for the Volume Algorithm applied to the Steiner Problem in Graphs, which is the basis algorithm for RVA and BVA, was jointly developed by the first author and Francisco Barahona from the IBM Research Center at Yorktown Heighs, NY, USA, see [1].

The preprocessed VLSI instances were provided by Eduardo Barboza, see [3].

## References

1. L. Bahiense, F. Barahona, and O. Porto. Solving Steiner Tree Problems in Graphs with Lagrangian Relaxation. Submitted to *Journal of Combinatorial Optimization*, 2000.
2. F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Math. Program.*, 87(3, Ser. A):385–399, 2000.
3. E. Barboza, M. V. Aragao and C. Ribeiro. Preprocessing Steiner Problems from VLSI Layout. Tech. Rep. MCC 32/99, Pontifícia Universidade Católica do Rio de Janeiro, 1999.
4. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Optimisation Numérique: aspects théoriques et pratiques.* Springer-Verlag, Berlin, 1997.
5. A. Claus and N. Maculan. Une nouvelle formulation du Problème de Steiner sur un graphe. Tech. Rep. 280, Centre de Recherche sur les Transports, Université de Montréal, 1983.
6. M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32:826–834, 1977.
7. J. B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms.* Number 305-306 in Grund. der math. Wiss. Springer-Verlag, Berlin, 1993.
8. F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 22:55–89, 1992.
9. F. K. Hwang, D. S. Richards, and P. Winter. The Steiner tree problem. In North Holland, editor, *Annals of Discrete Mathematics*, volume 53, 1992.

10. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
11. K. Knopp. *Infinite sequences and series.* Dover Publications Inc., New York, 1956.
12. T. Koch and A. Martin. Solving Steiner Tree Problems in Graphs to Optimality. Preprint SC-96-42, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, 1996.
13. G.M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.
14. C. Lemaréchal. An extension of Davidon methods to nondifferentiable problems. *Mathematical Programming Study*, 3:95–109, 1975.
15. N. Maculan. The Steiner Problem in Graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.
16. N. Shor. *Minimization methods for non-differentiable functions.* Springer-Verlag, Berlin, 1985.
17. M.V. Solodov and B.F. Svaiter. A hybrid approximate extragradient–proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Analysis*, 7:323–345, 1999.
18. H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Japon.*, 24(6):573–577, 1979/80.
19. P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38:431–446, 2000.
20. P. Winter. Steiner problems in networks: a survey. *Networks*, 17:129–167, 1987.
21. P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Study*, 3:145–173, 1975.
22. R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28:271–287, 1984.

## Appendix

We give here the complete tables with primal and dual results, for each preprocessed instance (see Table 1) and for each algorithm. Tables 5, 6, and 7, correspond, respectively, to VA, RVA, and BVA. A superscript $^\star$ in the column UB indicates that optimality was found.

Table 8 shows the additional costs that must be added to the optimal costs obtained with the preprocessed instances of Table 1 in order to recover the original optimal costs (before preprocessing).

| *Name* | LB | UB | *PrAp* | *SubPb* | (%)*Ser.* | *BestH* | *StopT* | *CPU(s)* |
|---|---|---|---|---|---|---|---|---|
| dmxa1721 | 172.15 | 173* | 239.94 | 420 | 9 | MSTV | (IDG) | 0.08 |
| dmxa1109 | 161.00 | 161* | 209.80 | 540 | 8 | MSTM | (IDG) | 0.39 |
| dmxa0903 | 419.95 | 420* | 463.18 | 1000 | 12 | MSTM | (IDG) | 8.18 |
| dmxa0848 | 424.10 | 425* | 516.12 | 760 | 12 | MSTM | (IDG) | 4.80 |
| dmxa1200 | 482.05 | 483* | 572.43 | 1060 | 21 | MSTM | (IDG) | 7.78 |
| dmxa0368 | 491.81 | 492* | 570.20 | 1040 | 11 | MSTM | (IDG) | 8.01 |
| dmxa1801 | 1205.12 | 1206* | 1309.82 | 1660 | 14 | MSTV | (IDG) | 320.14 |
| taq0631 | 219.21 | 220* | 277.33 | 720 | 11 | MSTM | (IDG) | 0.52 |
| taq0023 | 384.93 | 385* | 525.11 | 820 | 14 | MSTM | (IDG) | 4.82 |
| taq0739 | 692.50 | 695 | 694.75 | 301205 | 1 | T&MV | (PDF) | 1771.18 |
| taq0431 | 674.22 | 675* | 804.70 | 960 | 11 | MSTV | (IDG) | 23.76 |
| taq0741 | 784.38 | 785* | 883.71 | 1300 | 12 | MSTM | (IDG) | 29.60 |
| taq0751 | 784.24 | 785* | 914.50 | 1360 | 28 | MSTM | (IDG) | 66.44 |
| taq0365 | 1743.04 | 1744* | 1838.91 | 4720 | 4 | T&MV | (IDG) | 2826.97 |
| alue2087 | 378.79 | 379* | 509.60 | 900 | 19 | T&MV | (IDG) | 9.88 |
| alue5067 | 1458.12 | 1459* | 1636.01 | 1440 | 16 | T&MV | (IDG) | 564.35 |
| gap1904 | 117.27 | 118* | 151.94 | 680 | 9 | MSTV | (IDG) | 0.41 |
| gap2740 | 385.23 | 386* | 462.18 | 880 | 10 | MSTM | (IDG) | 0.93 |
| gap3100 | 488.49 | 489* | 624.20 | 720 | 30 | T&MV | (IDG) | 2.52 |
| gap3036 | 310.13 | 311* | 356.41 | 1060 | 22 | MSTV | (IDG) | 6.48 |
| gap2007 | 488.00 | 493 | 513.53 | 500000 | 1 | MSTM | !! | 147.86 |
| msm1931 | 417.00 | 417* | 599.35 | 460 | 16 | MSTV | (IDG) | 0.13 |
| msm2705 | 98.79 | 99* | 148.05 | 360 | 11 | MSTV | (IDG) | 0.03 |
| msm1844 | 65.42 | 66* | 100.01 | 460 | 16 | T&MV | (IDG) | 0.33 |
| msm0580 | 99.66 | 100* | 131.75 | 440 | 10 | MSTM | (IDG) | 0.23 |
| msm0920 | 131.24 | 132* | 171.53 | 600 | 19 | MSTV | (IDG) | 0.51 |
| msm1234 | 163.43 | 164* | 213.72 | 640 | 12 | MSTV | (IDG) | 0.34 |
| msm2802 | 120.71 | 121* | 191.16 | 460 | 14 | MSTM | (IDG) | 0.22 |
| msm2326 | 112.37 | 113* | 160.93 | 580 | 22 | MSTV | (IDG) | 0.73 |
| msm2525 | 160.47 | 161* | 201.75 | 860 | 20 | MSTV | (IDG) | 1.30 |
| msm1477 | 143.14 | 144* | 196.48 | 660 | 26 | T&MV | (IDG) | 1.23 |
| msm1008 | 244.42 | 245* | 316.16 | 640 | 24 | MSTV | (IDG) | 1.31 |
| msm4515 | 434.62 | 435* | 571.31 | 680 | 16 | MSTV | (IDG) | 4.04 |
| msm2492 | 519.50 | 522 | 542.43 | 500000 | 1 | T&MV | !! | 135.25 |
| msm2601 | 998.28 | 999* | 1127.85 | 1260 | 6 | T&MV | (IDG) | 50.29 |
| msm3829 | 1140.08 | 1141* | 1149.35 | 1660 | 6 | MSTV | (IDG) | 134.94 |
| msm2152 | 1145.29 | 1146* | 1299.89 | 1660 | 16 | MSTM | (IDG) | 230.82 |
| msm4312 | 1850.91 | 1887 | 1990.06 | 13760 | 1 | T&MV | !! | 7200.00 |
| msm2846 | 2185.17 | 2192 | 2266.18 | 33620 | 1 | MSTM | !! | 7200.00 |
| diw0487 | 83.47 | 84* | 116.39 | 520 | 10 | MSTV | (IDG) | 0.17 |
| diw0473 | 226.14 | 227* | 284.49 | 720 | 18 | MSTV | (IDG) | 1.35 |
| diw0459 | 326.32 | 327* | 390.94 | 840 | 12 | MSTV | (IDG) | 2.25 |
| diw0445 | 405.15 | 406* | 481.49 | 1020 | 13 | MSTM | (IDG) | 4.72 |
| diw0559 | 872.32 | 873* | 967.97 | 1380 | 13 | MSTM | (IDG) | 26.92 |
| diw0795 | 1377.24 | 1378* | 1535.27 | 1420 | 8 | T&MV | (IDG) | 88.35 |
| diw0778 | 1272.22 | 1273* | 1400.51 | 1420 | 12 | MSTV | (IDG) | 88.16 |
| diw0801 | 1480.29 | 1481* | 1593.87 | 1740 | 11 | T&MV | (IDG) | 200.70 |
| diw0234 | 1753.25 | 1754* | 2257.63 | 1040 | 10 | MSTM | (IDG) | 431.38 |
| diw0819 | 3044.33 | 3097 | 3314.21 | 6280 | 2 | MSTM | !! | 7200.00 |
| diw0820 | 3404.60 | 3809 | 4202.29 | 2100 | 11 | T&MV | !! | 7200.00 |

**Table 5.** Volume Algorithm (VA)

| *Name* | LB | UB | *PrAp* | *SubPb* | (%)*Ser.* | *BestH* | *StopT* | *CPU(s)* |
|---|---|---|---|---|---|---|---|---|
| dmxa1721 | 172.13 | 173* | 222.35 | 573 | 7 | MSTV | (IDG) | 0.25 |
| dmxa1109 | 160.09 | 161* | 211.14 | 436 | 12 | MSTM | (IDG) | 0.41 |
| dmxa0903 | 419.07 | 420* | 478.73 | 964 | 19 | T&MV | (IDG) | 8.56 |
| dmxa0848 | 424.17 | 425* | 508.88 | 774 | 20 | MSTM | (IDG) | 5.24 |
| dmxa1200 | 482.20 | 483* | 552.74 | 1072 | 26 | MSTV | (IDG) | 7.96 |
| dmxa0368 | 491.39 | 492* | 560.88 | 1042 | 21 | MSTM | (IDG) | 8.57 |
| dmxa1801 | 1205.15 | 1206* | 1291.38 | 2035 | 25 | MSTV | (IDG) | 338.20 |
| taq0631 | 219.42 | 220* | 259.07 | 845 | 13 | MSTM | (IDG) | 0.62 |
| taq0023 | 384.78 | 385* | 474.01 | 803 | 22 | MSTM | (IDG) | 4.99 |
| taq0739 | 692.47 | 695 | 693.32 | 250449 | 13 | T&MV | (45) | 2123.38 |
| taq0431 | 674.33 | 675* | 820.68 | 1408 | 18 | MSTV | (IDG) | 29.22 |
| taq0741 | 784.17 | 785* | 867.85 | 1399 | 18 | MSTM | (IDG) | 30.64 |
| taq0751 | 784.31 | 785* | 911.03 | 1426 | 35 | MSTV | (IDG) | 69.90 |
| taq0365 | 1743.31 | 1744* | 1972.46 | 11875 | 19 | T&MV | (IDG) | 6835.95 |
| alue2087 | 378.23 | 379* | 491.89 | 767 | 29 | T&MV | (IDG) | 7.02 |
| alue5067 | 1458.32 | 1459* | 1558.89 | 2003 | 26 | T&MV | (IDG) | 405.69 |
| gap1904 | 117.11 | 118* | 158.08 | 648 | 13 | MSTV | (IDG) | 0.36 |
| gap2740 | 385.52 | 386* | 476.25 | 836 | 16 | MSTV | (IDG) | 1.02 |
| gap3100 | 488.64 | 489* | 592.93 | 840 | 31 | T&MV | (IDG) | 2.77 |
| gap3036 | 310.16 | 311* | 349.33 | 1006 | 27 | T&MV | (IDG) | 5.65 |
| gap2007 | 487.89 | 493 | 511.49 | 500000 | 18 | MSTV | !! | 230.76 |
| msm1931 | 417.00 | 417* | 532.19 | 691 | 21 | MSTV | (IDG) | 0.30 |
| msm2705 | 98.94 | 99* | 144.64 | 441 | 12 | MSTV | (IDG) | 0.12 |
| msm1844 | 65.32 | 66* | 93.48 | 423 | 13 | MSTV | (IDG) | 0.25 |
| msm0580 | 99.13 | 100* | 128.74 | 413 | 13 | MSTM | (IDG) | 0.26 |
| msm0920 | 131.08 | 132* | 171.86 | 710 | 18 | MSTV | (IDG) | 0.44 |
| msm1234 | 163.37 | 164* | 228.65 | 486 | 14 | MSTV | (IDG) | 0.25 |
| msm2802 | 120.04 | 121* | 179.23 | 589 | 12 | MSTM | (IDG) | 0.30 |
| msm2326 | 112.13 | 113* | 162.85 | 639 | 22 | MSTV | (IDG) | 0.65 |
| msm2525 | 160.20 | 161* | 214.54 | 809 | 22 | MSTV | (IDG) | 1.16 |
| msm1477 | 143.04 | 144* | 186.47 | 781 | 24 | T&MV | (IDG) | 1.20 |
| msm1008 | 244.43 | 245* | 317.31 | 816 | 24 | MSTV | (IDG) | 1.60 |
| msm4515 | 434.16 | 435* | 573.04 | 597 | 26 | MSTV | (IDG) | 3.78 |
| msm2492 | 519.42 | 522 | 538.56 | 500000 | 25 | T&MV | !! | 241.73 |
| msm2601 | 998.23 | 999* | 1171.62 | 1570 | 16 | MSTV | (IDG) | 62.84 |
| msm3829 | 1140.75 | 1141* | 1147.81 | 9156 | 24 | MSTV | (IDG) | 561.36 |
| msm2152 | 1145.36 | 1146* | 1199.42 | 2008 | 25 | MSTV | (IDG) | 261.51 |
| msm4312 | 1852.66 | 1893 | 1951.36 | 11320 | 18 | T&MV | !! | 7200.00 |
| msm2846 | 2186.10 | 2191 | 2315.53 | 25820 | 20 | T&MV | !! | 7200.00 |
| diw0487 | 83.20 | 84* | 109.26 | 418 | 10 | MSTV | (IDG) | 0.08 |
| diw0473 | 226.24 | 227* | 291.73 | 734 | 21 | MSTV | (IDG) | 1.08 |
| diw0459 | 326.02 | 327* | 379.63 | 900 | 18 | MSTV | (IDG) | 2.12 |
| diw0445 | 405.08 | 406* | 452.54 | 820 | 22 | T&MV | (IDG) | 3.44 |
| diw0559 | 872.10 | 873* | 930.41 | 1502 | 21 | MSTM | (IDG) | 22.87 |
| diw0795 | 1377.32 | 1378* | 1500.89 | 1710 | 23 | T&MV | (IDG) | 109.82 |
| diw0778 | 1272.54 | 1273* | 1381.91 | 2008 | 24 | MSTV | (IDG) | 97.34 |
| diw0801 | 1480.22 | 1481* | 1596.24 | 6744 | 17 | MSTV | (IDG) | 525.96 |
| diw0234 | 1753.33 | 1754* | 2042.07 | 1687 | 25 | T&MV | (IDG) | 410.18 |
| diw0819 | 3024.48 | 3115 | 3620.67 | 5100 | 20 | MSTV | !! | 7200.00 |
| diw0820 | 3081.41 | 3833 | 6634.23 | 1880 | 25 | T&MV | !! | 7200.00 |

**Table 6.** Revised Volume Algorithm (RVA)

| *Name* | LB | UB | *PrAp* | *SubPb* | (%)*Ser.* | *BestH* | *StopT* | *CPU(s)* |
|---|---|---|---|---|---|---|---|---|
| dmxa1721 | 172.14 | 173* | 241.64 | 411 | 10 | MSTV | (IDG) | 0.16 |
| dmxa1109 | 160.10 | 161* | 215.53 | 474 | 12 | MSTM | (IDG) | 0.50 |
| dmxa0903 | 419.11 | 420* | 491.78 | 849 | 23 | T&MV | (IDG) | 8.34 |
| dmxa0848 | 424.42 | 425* | 512.47 | 768 | 20 | MSTM | (IDG) | 5.15 |
| dmxa1200 | 482.11 | 483* | 561.40 | 1030 | 30 | MSTV | (IDG) | 8.04 |
| dmxa0368 | 491.94 | 492* | 564.91 | 1018 | 18 | MSTM | (IDG) | 7.90 |
| dmxa1801 | 1205.05 | 1206* | 1294.03 | 1747 | 25 | MSTV | (IDG) | 317.59 |
| taq0631 | 219.27 | 220* | 272.01 | 799 | 13 | MSTM | (IDG) | 0.56 |
| taq0023 | 384.86 | 385* | 514.81 | 856 | 20 | T&MV | (IDG) | 4.86 |
| taq0739 | 692.45 | 695 | 697.68 | 500000 | 1 | T&MV | !! | 3071.25 |
| taq0431 | 674.18 | 675* | 835.81 | 1164 | 18 | MSTV | (IDG) | 25.78 |
| taq0741 | 784.52 | 785* | 883.44 | 1414 | 18 | T&MV | (IDG) | 31.92 |
| taq0751 | 784.06 | 785* | 934.19 | 1274 | 38 | MSTV | (IDG) | 69.39 |
| taq0365 | 1732.62 | 1744 | 1841.82 | 26420 | 3 | T&MV | !! | 7200.00 |
| alue2087 | 378.73 | 379* | 498.06 | 766 | 31 | T&MV | (IDG) | 9.43 |
| alue5067 | 1458.05 | 1459* | 1628.90 | 1671 | 24 | MSTM | (IDG) | 709.75 |
| gap1904 | 117.19 | 118* | 152.33 | 680 | 13 | MSTV | (IDG) | 0.45 |
| gap2740 | 385.21 | 386* | 479.01 | 727 | 17 | MSTV | (IDG) | 1.13 |
| gap3100 | 488.45 | 489* | 606.91 | 769 | 34 | T&MV | (IDG) | 2.77 |
| gap3036 | 310.14 | 311* | 364.67 | 1000 | 30 | MSTM | (IDG) | 6.14 |
| gap2007 | 488.00 | 493 | 496.25 | 500000 | 1 | MSTV | !! | 1543.56 |
| msm1931 | 417.00 | 417* | 599.34 | 460 | 19 | MSTV | (IDG) | 0.19 |
| msm2705 | 98.79 | 99* | 149.46 | 342 | 14 | MSTV | (IDG) | 0.12 |
| msm1844 | 65.26 | 66* | 99.17 | 472 | 19 | MSTV | (IDG) | 0.36 |
| msm0580 | 99.47 | 100* | 132.06 | 435 | 14 | MSTM | (IDG) | 0.34 |
| msm0920 | 131.28 | 132* | 166.56 | 661 | 21 | MSTV | (IDG) | 0.62 |
| msm1234 | 163.64 | 164* | 216.34 | 603 | 15 | MSTV | (IDG) | 0.47 |
| msm2802 | 120.42 | 121* | 192.90 | 448 | 17 | MSTM | (IDG) | 0.36 |
| msm2326 | 112.44 | 113* | 161.45 | 574 | 26 | MSTV | (IDG) | 0.79 |
| msm2525 | 160.19 | 161* | 203.64 | 836 | 25 | MSTV | (IDG) | 1.61 |
| msm1477 | 143.61 | 144* | 184.75 | 795 | 26 | T&MV | (IDG) | 1.45 |
| msm1008 | 244.51 | 245* | 317.56 | 633 | 30 | MSTV | (IDG) | 1.44 |
| msm4515 | 434.09 | 435* | 577.85 | 648 | 23 | MSTV | (IDG) | 3.77 |
| msm2492 | 519.40 | 522 | 593.55 | 500000 | 1 | T&MV | !! | 1356.71 |
| msm2601 | 998.22 | 999* | 1114.70 | 1277 | 17 | MSTV | (IDG) | 53.80 |
| msm3829 | 1140.96 | 1141* | 1170.60 | 18140 | 16 | MSTM | (IDG) | 895.31 |
| msm2152 | 1145.09 | 1146* | 1266.43 | 1634 | 26 | MSTM | (IDG) | 234.17 |
| msm4312 | 1794.09 | 1893 | 1981.25 | 39020 | 2 | T&MV | !! | 7200.00 |
| msm2846 | 2174.43 | 2192 | 2246.70 | 35020 | 1 | T&MV | !! | 7200.00 |
| diw0487 | 83.39 | 84* | 113.13 | 580 | 11 | MSTV | (IDG) | 0.17 |
| diw0473 | 226.10 | 227* | 292.29 | 642 | 24 | MSTV | (IDG) | 1.19 |
| diw0459 | 326.02 | 327* | 383.37 | 895 | 17 | MSTV | (IDG) | 2.30 |
| diw0445 | 405.24 | 406* | 449.98 | 1039 | 20 | MSTM | (IDG) | 4.78 |
| diw0559 | 872.20 | 873* | 965.39 | 1366 | 22 | MSTV | (IDG) | 30.33 |
| diw0795 | 1377.08 | 1378* | 1553.04 | 1316 | 21 | T&MV | (IDG) | 84.85 |
| diw0778 | 1272.02 | 1273* | 1406.10 | 1305 | 22 | MSTV | (IDG) | 92.35 |
| diw0801 | 1480.08 | 1481* | 1581.98 | 1996 | 21 | MSTV | (IDG) | 211.82 |
| diw0234 | 1753.35 | 1754* | 2258.90 | 1522 | 20 | MSTM | (IDG) | 609.02 |
| diw0819 | 3041.46 | 3097 | 3243.44 | 6060 | 17 | MSTV | !! | 7200.00 |
| diw0820 | 3181.67 | 3833 | 5493.79 | 1880 | 23 | T&MV | !! | 7200.00 |

**Table 7.** Bundle Version of the Revised Volume Algorithm (BVA)

| *Name* | $cost_{add}$ |
|---|---|
| dmxa1721 | 607 |
| dmxa1109 | 293 |
| dmxa0903 | 160 |
| dmxa0848 | 169 |
| dmxa1200 | 267 |
| dmxa0368 | 525 |
| dmxa1801 | 159 |
| taq0631 | 361 |
| taq0023 | 236 |
| taq0739 | 153 |
| taq0431 | 222 |
| taq0741 | 62 |
| taq0751 | 154 |
| taq0365 | 170 |
| alue2087 | 670 |
| alue5067 | 1127 |
| gap1904 | 645 |
| gap2740 | 359 |
| gap3100 | 151 |
| gap3036 | 146 |
| gap2007 | 611 |
| msm1931 | 187 |
| msm2705 | 615 |
| msm1844 | 122 |
| msm0580 | 367 |
| msm0920 | 674 |
| msm1234 | 386 |
| msm2802 | 805 |
| msm2326 | 286 |
| msm2525 | 1129 |
| msm1477 | 924 |
| msm1008 | 249 |
| msm4515 | 195 |
| msm2492 | 937 |
| msm2601 | 441 |
| msm3829 | 430 |
| msm2152 | 444 |
| msm4312 | 149 |
| msm2846 | 944 |
| diw0487 | 1340 |
| diw0473 | 871 |
| diw0459 | 1035 |
| diw0445 | 957 |
| diw0559 | 697 |
| diw0795 | 172 |
| diw0778 | 900 |
| diw0801 | 106 |
| diw0234 | 242 |
| diw0819 | 302 |
| diw0820 | 428 |

**Table 8.** Additional costs to recover the VLSI Instances after preprocessing