

A Randomized Algorithm for the MaxFS Problem

Edoardo Amaldi §, Pietro Belotti §, and Raphael Hauser †

§ DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy

{amaldi,belotti}@elet.polimi.it

† Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, UK

hauser@comlab.ox.ac.uk

1 Introduction

We consider the following problem, referred to as MAXFS:

Given an infeasible linear system $\mathbf{Ax} \geq \mathbf{b}$ with $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, find a Feasible Subsystem containing as many inequalities as possible.

This problem has a number of interesting applications in various fields such as operations research [7, 12], radiation therapy [13], image and signal processing [15], statistical discriminant analysis and machine learning [4, 14]. The growing interest for MAXFS is due to the fact that many complex phenomena that can be well approximated with linear models yield formulations involving large and generally infeasible linear systems for which approximate solutions in terms of l_1 or l_2 norms are not meaningful. Note that, since linear system feasibility can be checked in polynomial time, the MAXFS structure differs substantially from that of MAX SAT, the well-known problem of satisfying a maximum number of Boolean clauses. Like MAX SAT though MAXFS can be approximated within a factor of 2 but it does not admit a polynomial-time approximation scheme, unless $P = NP$ [2].

Besides the large-scale instances arising in classification problems [4, 14] and when coping with infeasibility in linear programming [7, 12], in several recent and challenging applications to telecommunications (digital video and audio broadcasting [21]) and to computational biology (modeling the energy function underlying the folding of proteins [16]) one faces very large MAXFS instances with tens of thousands up to millions of inequalities in hundreds up to thousands of variables. Such instances are beyond the reach of available computational approaches, including state-of-the-art MIP solvers applied to big-M formulations, the best available heuristics [7] and the latest polyhedral techniques based on partial set covering formulations [19, 3] and combinatorial Benders' cuts [8].

To tackle large-scale instances of MAXFS, we consider randomized and thermal variants of the classical Agmon-Motzkin-Schoenberg (AMS) relaxation method for solving systems of linear inequalities. Deterministic AMS relaxation versions have been extensively studied in the mathematical programming literature, often as special cases of subgradient methods (see e.g. [11, 22]), and in the machine learning literature under the name of perceptron procedures (see e.g. [5, 17]). The algorithms we propose and investigate here extend the thermal perceptron heuristic [10]. It is worth pointing out that the power of randomization in relaxation/subgradient type methods, which was first exploited by the machine learning community, has also been recently studied in [20] for *feasible* systems of convex inequalities and in [18] for minimizing a *convex* function that consists of a sum of a large number of convex component functions. Here the focus is on a difficult combinatorial optimization problem.

After describing the randomized and thermal relaxation methods, we establish lower bounds on the probability that they yield an optimal solution of MAXFS within a given number of iterations as a function of the input data. These bounds imply finite termination to an optimal solution with probability one. To show that these methods do also perform very well in practice, we report in Section 4 some computational results obtained for large-scale instances arising from digital video broadcasting and modeling protein folding potentials.

2 Randomized and thermal relaxation methods

The *relaxation method* is a simple algorithmic framework for solving systems of linear inequalities, see e.g. [11, 22]. The idea is to iteratively relax all but one violated inequality and to update the current approximate solution by moving along the orthogonal direction to the corresponding hyperplane until a feasible solution is found. Given a feasible system $A\mathbf{x} \geq \mathbf{b}$ with $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, and a starting point $\mathbf{x}_0 \in \mathbb{R}^n$, the method generates a sequence of iterates $(\mathbf{x}_i)_{\mathbb{N}} \subset \mathbb{R}^n$ as follows: at iteration i an index $k_i \in \{1, \dots, m\}$ is selected according to some specified rule, the corresponding inequality $\mathbf{a}_{k_i}\mathbf{x} \geq b_{k_i}$ is considered, and the next iterate is determined via

$$\mathbf{x}_{i+1} = \begin{cases} \mathbf{x}_i + \eta_i \mathbf{a}_{k_i} & \text{with probability } p_i \text{ if } \mathbf{a}_{k_i}\mathbf{x}_i < b_{k_i} \\ \mathbf{x}_i & \text{otherwise} \end{cases} \quad (2.1)$$

where $\eta_i > 0$ and $p_i \in [0, 1]$ may be chosen differently in every iteration. When $\mathbf{x}_{i+1} \neq \mathbf{x}_i$ we say that an update has occurred.

Let us briefly explain the motivation behind the parameters η_i and p_i . The *step length multipliers* $\eta_i > 0$ provide a tool to account for varying degrees of urgency to "fix" the violation of the selected inequality. For example, in the cyclic projection method $\eta_i = (b_{k_i} - \mathbf{a}_{k_i}\mathbf{x}_i) / \|\mathbf{a}_{k_i}\|$ is chosen so that $\mathbf{a}_{k_i}\mathbf{x}_{i+1} = b_{k_i}$. In classical versions, the inequalities are considered in cyclic order or at each iteration one of those with largest violation is selected. In the *randomized* relaxation method, we assume that k_i is chosen uniformly at random. We also consider a second level of randomization by introducing the *updating probability* p_i that an update is actually carried out. When $p_i < 1$, we speak of a *doubly randomized* method, while $p_i \equiv 1$ corresponds to the standard (singly) randomized relaxation method. In the MAXFS context where we apply variants of these methods to infeasible systems, double randomization can avoid roundoff problems by making it possible to replace a large number of small updates by updates of moderate size that occur with small probability.

To look for maximum feasible subsystems of infeasible systems $A\mathbf{x} \geq \mathbf{b}$, we consider *thermal* variants of the randomized relaxation method (RTR). In these variants the choice of the step-length multipliers η_i aims at favoring updates that are due to unsatisfied inequalities with a relatively small *violation* $v_i = \max\{0, b_{k_i} - \mathbf{a}_{k_i}\mathbf{x}_i\}$. The rationale is that, if previous iterations led \mathbf{x}_i in a region of the solution space containing good approximate solutions of the MAXFS problem, then \mathbf{x}_i should be improved by small local updates. The large modifications of \mathbf{x}_i that would be required when v_i is large could indeed destroy much of the progress that has previously been achieved. A natural way to gradually reduce the attention paid to unsatisfied inequalities with large violations is to introduce a decreasing *temperature schedule* $(t_i)_{\mathbb{N}} \subset \mathbb{R}_+$ and to choose $(\eta_i)_{\mathbb{N}}$ as a function of t_i and v_i , for example, by setting:

$$\eta_i = \frac{t_i}{t_0} \exp(-v_i/t_i). \quad (2.2)$$

For large values of t_i , any violated inequality yields a significant update, while for low values, only those with small violations with respect to t_i yield significant updates. Like in simulated

annealing (SA) t_i can be seen as a temperature that gradually “cools down” the activity level of the algorithm and forces it to pursue more localized searches. The similarity, however, stops at this intuitive level and the algorithms differ in important ways. In the doubly randomized RTR variant, which seems to resemble SA, the updates are carried out with exponentially decreasing probability irrespectively of whether the overall objective function improves or worsens. The analysis of Section 3 is based on ideas that do not appear in the SA literature and our probabilistic finite termination results are much stronger than SA asymptotic convergence guarantees.

When RTR variants are applied to infeasible systems, the best iterate \mathbf{x} found up to iteration i needs to be stored in a so-called *ratchet vector* \mathbf{z}_i . In practical implementations the ratchet vector obtained after a predetermined maximum number of iterations is returned as an approximate solution of MAXFS.

In summary, different RTR variants are characterized by:

- the randomized *selection rule* which specifies the way a single inequality is selected at each iteration (e.g., uniformly at random with or without replacement),
- the *step-length multiplier rule* which specifies how the sequence of η_i is determined (e.g., η_i is constant or defined as in (2.2) while t_i is linearly/exponentially reduced from t_0 to 0),
- the *decision rule* which is characterized by the probability p_i , $0 \leq p_i \leq 1$, that the update at iteration i is actually carried out.

It is worth pointing out that RTR methods can be naturally viewed as randomized sub-gradient homotopy method in which subgradients are computed incrementally. The inherent parallelism of these methods make them well-suited to tackle even very large MAXFS instances such as those arising, for example, when designing protein folding potentials [16]. See [6] for parallel versions of relaxation methods for linear (convex) feasible systems.

3 Probabilistic termination guarantees

In this section we give an example of the type of probabilistic termination guarantee to an optimal solution of MAXFS that we established for several RTR variants. In this extended abstract we will focus on the *doubly* randomized RTR variant with *constant* η_i applied to homogeneous systems with strict inequalities $A\mathbf{x} > \mathbf{0}$. Relevant applications and computational results will be discussed in Section 4. W.l.o.g. we may assume that all row vectors \mathbf{a}_k of A have unit length. By abuse of language we write $\mathbf{a} \in A$ to indicate that \mathbf{a} is one of the row vectors of A , and we write \mathbf{a}_i for the row vector that corresponds to the inequality selected at iteration i . For the purposes of clarity, let us now summarize the algorithm we intend to analyze.

Algorithm 1. Let $(t_i)_{\mathbb{N}}$ be a decreasing temperature schedule, $\mathbf{x}_0 \in \mathbb{R}^n$ a starting point with unit norm, and $\mathbf{z}_0 := \mathbf{x}_0$. Let $(\mathbf{a}_i)_{\mathbb{N}}$ be i.i.d. random vectors with uniform distribution over A . Then the sequence $(U_i)_{\mathbb{N}}$ of binary random variables that formalize the updates and the sequences $(\mathbf{x}_i)_{\mathbb{N}}$ and $(\mathbf{z}_i)_{\mathbb{N}}$ of random vectors in \mathbb{R}^n are evaluated via the following recursive relations:

- i) U_i is $\sigma(\mathbf{a}_i, \mathbf{x}_i)$ -measurable and such that

$$p_i := \mathbb{P}[U_i = 1 \mid \mathbf{a}_i \cdot \mathbf{x}_i \leq 0] = \frac{t_i}{t_0} \exp\left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i}\right), \quad \mathbb{P}[U_i = 0 \mid \mathbf{a}_i \cdot \mathbf{x}_i \leq 0] = 1 - p_i,$$

$$\mathbb{P}[U_i = 1 \mid \mathbf{a}_i \cdot \mathbf{x}_i > 0] = 0 \quad \text{and} \quad \mathbb{P}[U_i = 0 \mid \mathbf{a}_i \cdot \mathbf{x}_i > 0] = 1,$$
- ii) $\mathbf{x}_{i+1} = \mathbf{x}_i + U_i \mathbf{a}_i,$

iii) $\mathbf{z}_{i+1} = \mathbf{x}_{i+1}$ if $\#A_{\mathbf{x}_{i+1}} > \#A_{\mathbf{z}_i}$, and $\mathbf{z}_{i+1} = \mathbf{z}_i$ otherwise.

Thus, the algorithm uses the step-length multiplier sequence $(\eta_i)_{\mathbb{N}} = (1)_{\mathbb{N}}$.

In Theorem 2 below we will bound the probability that the ratchet vector \mathbf{z}_i becomes MAXFS-optimal in at most i iterations, and we will establish that Algorithm 1 almost surely finds an optimal solution in finite number of iterations. Before we formulate this result, let us outline the general line of attack we adopt for its analysis.

For any $\mathbf{x} \in \mathbb{R}^n$ let $A_{\mathbf{x}} := \{\mathbf{a} \in A : \mathbf{a} \cdot \mathbf{x} > 0\}$ denote the coefficient vectors of the subset of inequalities satisfied by \mathbf{x} and $\#A_{\mathbf{x}}$ its cardinality. Let $\tilde{\mathbf{x}}$ be an optimal solution of the MAXFS problem, that is, $\#A_{\tilde{\mathbf{x}}} = m^* := \max\{\#A_{\mathbf{x}} : \mathbf{x} \in \mathbb{R}^n\}$, and let $\mathbf{x}^* := \tilde{\mathbf{x}}/\alpha$, where $\alpha := \min\{\mathbf{a} \cdot \tilde{\mathbf{x}} : a \in A_{\tilde{\mathbf{x}}}\}$. Let us now focus our attention on an optimal solution \mathbf{x}^* and the associated feasible subsystem $A_{\mathbf{x}^*}\mathbf{x} > \mathbf{0}$. As long as the iterate \mathbf{x}_i satisfies $\#A_{\mathbf{x}_i} < m^*$, there exists at least one $\mathbf{a} \in A_{\mathbf{x}^*}$ such that $\mathbf{a} \cdot \mathbf{x}_i \leq 0$. Under uniform sampling this particular inequality is drawn with probability $1/m$, and then $\mathbf{x}_{i+1} \neq \mathbf{x}_i$ with probability p_i . According to Lemma 3 below no more than a certain number $\tilde{\gamma}$ of consecutive such updates can occur before the algorithm finds an \mathbf{x}_i such that $\mathbf{a} \cdot \mathbf{x}_i > 0$ for all $\mathbf{a} \in A_{\mathbf{x}^*}$, and hence \mathbf{x}_i is optimal for MAXFS. Subdividing the sequence of iterates into epochs of length $\tilde{\gamma}$, optimality is achieved with positive probability in each epoch. Although the law of large numbers can be seen as the motivating idea of our approach, the analysis is more complicated because the events that optimality is achieved in iteration i are not independent for consecutive iterations.

The convergence speed claimed in Theorem 2 will be expressed as a function of the condition number $\kappa(A)$ that we defined in [1] as the inverse of the smallest strictly positive k -dimensional volume of the parallelepiped generated by any subset of k row vectors of A for all $k = 1, \dots, m$. In the case where A consists of rational data of total bit length \mathcal{L} , the inequality $\kappa(A) \leq 2^{\mathcal{L}}$ holds, see [1] for details.

Theorem 2. *Let $M := 2\kappa(A)2^{3n/2} + 1$ and $\gamma := (M + \|\mathbf{x}^*\|)^2$. If the temperature schedule of Algorithm 1 satisfies $t_i \geq \frac{c}{\log i}$ for $i \geq 2$ and $t_i \geq \frac{c}{\log 2}$ for $i = 0, 1$ for some constant $c \geq 2M\gamma$, then*

i) *for all $k \in \mathbb{N}$, we have: $\mathbb{P}[\#A_{\mathbf{z}_{k\gamma}} = m^*] \geq 1 - \prod_{j=2+\lfloor \frac{j}{\tilde{\gamma}} \rfloor}^k \left(1 - \frac{1}{j\tilde{\gamma}}\right)$ where $r > 0$ is large enough for $\frac{m \log x}{\log 2} < x^{\frac{1}{2\tilde{\gamma}}}$ to hold for all $x > r$.*

ii) *almost surely the ratchet vector becomes MAXFS-optimal after a finite number of iterations.*

Before we can prove Theorem 2, we need to introduce two lemmas. The first result deals with the finite termination of the relaxation method on feasible problem instances.

Lemma 3. *Let (any variant of) the relaxation method be applied to a feasible linear system $A\mathbf{x} > \mathbf{0}$ and let $(i(l))_{\mathbb{N}} \subset \mathbb{N}$ be the sequence of iterations in which updates occur. If $\lim_{u \rightarrow \infty} \sum_{l=1}^u \eta_{i(l)}(2 - \eta_{i(l)}) = +\infty$ then $\mathbf{x}_{i(l)}$ becomes a feasible solution for some $l \leq \tilde{\gamma} := \max\left\{u \in \mathbb{N} : \sum_{l=1}^u \eta_{i(l)}(2 - \eta_{i(l)}) \leq (\|\mathbf{x}_0\| + \|\mathbf{x}^*\|)^2\right\}$.*

Note that $\tilde{\gamma} = \tilde{\gamma}(\|\mathbf{x}_0\|, \|\mathbf{x}^*\|, A, (\eta_{i(l)})_{\mathbb{N}})$ is a monotone increasing function of $\|\mathbf{x}_0\|$ if all other entries are fixed. Lemma 3 can easily be shown by adapting the proof of a similar result from [5, 17] that was given for constant $\eta_i \equiv 1$. It has long been known that the number of updates required can grow exponentially in the input size [22], but we note that a variant of the relaxation method in which the underlying space is periodically rescaled was recently shown to be polynomial in a probabilistic framework [9].

In the case of infeasible systems, infinitely many updates occur due to violated inequalities. However, our second lemma shows the interesting fact that under bounded step-length multipliers the sequence of iterates $(\mathbf{x}_i)_{\mathbb{N}}$ remains bounded.

Lemma 4. *Let $A\mathbf{x} > 0$ be an arbitrary homogeneous system, $\mathbf{x}_0 \in \mathbb{R}^n$ and $(\mathbf{x}_i)_{i \in \mathbb{N}}$ the sequence of iterates generated by any version of the relaxation method that uses a bounded sequence of step-length multipliers $0 < \eta_i \leq \bar{\eta}$ for some $\bar{\eta} > 0$. Then for all i , we have*

$$\|\mathbf{x}_i\| \leq M(A, \mathbf{x}_0) := 2 \max \left\{ \|\mathbf{x}_0\|, \bar{\eta} (\kappa(A) 2^{3n/2} + 1) \right\}.$$

For a proof see Theorem 6.1 in [1]. Lemma 4 is a generalization of the classical *perceptron boundedness theorem* [5, 17] to non-constant step-length multipliers, but its proof is based on a different approach in order to settle the open question of expressing M as an explicit function of the input data (A, \mathbf{x}_0) . This explicit expression is a crucial ingredient in our proof of Theorem 2, which we are now ready to present.

Proof. (Theorem 2) Under the assumptions $\|\mathbf{a}\| = 1 \forall \mathbf{a} \in A$ and $\|\mathbf{x}_0\| = 1$, Lemma 4 implies that $\|\mathbf{x}_i\| \leq M$ for all $i \in \mathbb{N}$. The above-mentioned monotonicity of $\tilde{\gamma}$ implies that

$$\tilde{\gamma}(\|\mathbf{x}_{k\gamma}\|, \|\mathbf{x}^*\|, A, (1)_{\mathbb{N}}) \leq \gamma, \quad \forall k \in \mathbb{N}. \quad (3.1)$$

Since $(1)_{\mathbb{N}}$ is shift-invariant, we can concentrate on epochs of constant length γ under the general framework outlined above.

i) Denoting the complement of a set $B \subseteq A$ by $B^c = A \setminus B$, we define the events $E_i := \{\#A_{\mathbf{x}_i} = m^*\} \cup \{\mathbf{a}_i \in A_{\mathbf{x}^*} \cap A_{\mathbf{x}_i}^c, U_i = 1\}$. We claim that for all $k \in \mathbb{N}$,

$$E^{[k]} := E_{k\gamma-1} \cap \dots \cap E_{(k-1)\gamma} \subseteq \{\#A_{\mathbf{z}_{k\gamma}} = m^*\}. \quad (3.2)$$

Indeed, suppose that $E^{[k]}$ occurs. If $\#A_{\mathbf{x}_i} = m^*$ for some index $i \in [(k-1)\gamma, k\gamma-1]$ then the ratchet will assure that $\#A_{\mathbf{z}_i} = m^*$ and a fortiori $\#A_{\mathbf{z}_{k\gamma}} = m^*$. Otherwise, in γ consecutive iterations, a violated inequality $\mathbf{a}_i \cdot \mathbf{x}_i \leq 0$, $\mathbf{a}_i \in A_{\mathbf{x}^*}$ is drawn and \mathbf{x}_i is updated to $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{a}_i$. But then (3.1) and Lemma 3 imply that $A_{\mathbf{x}_i} = A_{\mathbf{x}^*}$ occurs for some $i \in [(k-1)\gamma + 1, k\gamma]$, and then it is again the case that $\#A_{\mathbf{z}_{k\gamma}} = m^*$, establishing the truth of our claim.

Equation (3.2) implies

$$\mathbb{P} \left[\#A_{\mathbf{z}_{k\gamma}} < m^* \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^* \right] \leq 1 - \mathbb{P} \left[E^{[k]} \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^* \right]. \quad (3.3)$$

Our next goal is to bound the right-hand side of (3.3). As long as $\#A_{\mathbf{x}_i} < m$, we have $A_{\mathbf{x}^*} \cap A_{\mathbf{x}_i}^c \neq \emptyset$, so that for any $\sigma(\mathbf{x}_0; \mathbf{a}_0, \dots, \mathbf{a}_{i-1}; U_0, \dots, U_{i-1})$ -measurable event $F \subseteq \{\#A_{\mathbf{x}_i} < m^*\}$ it is true that

$$\mathbb{P} [E_i \mid F] = \mathbb{P} [\mathbf{a}_i \in A_{\mathbf{x}^*} \cap A_{\mathbf{x}_i}^c, U_i = 1 \mid F] \geq \frac{1}{m} \times \frac{t_i}{t_0} \exp \left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i} \right). \quad (3.4)$$

In particular, $B_i \cap \{\#A_{\mathbf{x}_i} < m^*\}$ is $\sigma(\mathbf{x}_0; \mathbf{a}_0, \dots, \mathbf{a}_{i-1}; U_0, \dots, U_{i-1})$ -measurable for $i > (k-1)\gamma$, where $B_i := \left(\bigcap_{j=(k-1)\gamma}^{i-1} E_j \right) \cap \left\{ \#A_{\mathbf{z}_{(k-1)\gamma}} < m^* \right\}$.

Therefore,

$$\begin{aligned} \mathbb{P} [E_i \mid B_i] &= \mathbb{P} [E_i \mid \#A_{\mathbf{x}_i} < m^*, B_i] \times \mathbb{P} [\#A_{\mathbf{x}_i} < m^* \mid B_i] \\ &\quad + \mathbb{P} [E_i \mid \#A_{\mathbf{x}_i} = m^*, B_i] \times \mathbb{P} [\#A_{\mathbf{x}_i} = m^* \mid B_i] \\ &\stackrel{(3.4)}{\geq} \frac{t_i}{mt_0} \exp \left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i} \right) \times \mathbb{P} [\#A_{\mathbf{x}_i} < m^* \mid B_i] + 1 \times \mathbb{P} [\#A_{\mathbf{x}_i} = m^* \mid B_i] \\ &\geq \frac{t_i}{mt_0} \exp \left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i} \right), \end{aligned}$$

so that

$$\begin{aligned} \mathbb{P}[E^{[k]} \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^*] &= \mathbb{P}[E_{(k-1)\gamma} \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^*] \times \prod_{i=(k-1)\gamma+1}^{k\gamma-1} \mathbb{P}[E_i \mid B_i] \\ &\geq \prod_{i=(k-1)\gamma}^{k\gamma-1} \frac{t_i}{mt_0} \exp\left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i}\right). \end{aligned} \quad (3.5)$$

We wish to bound this quantity solely in terms of the epoch k . The assumption on the temperature schedule implies $\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i} < \frac{M \log i}{2M\gamma} = \frac{\log i}{2\gamma}$, and hence,

$$\frac{t_i}{mt_0} \exp\left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i}\right) > \frac{\log 2}{m \log i} \times i^{-\frac{1}{2\gamma}}.$$

Let $r > 0$ be such that $\frac{m \log x}{\log 2} < x^{\frac{1}{2\gamma}}$ for all $x > r$. Then for $i > r$, $\frac{t_i}{mt_0} \exp\left(-\frac{|\mathbf{a}_i \cdot \mathbf{x}_i|}{t_i}\right) > i^{-\frac{1}{\gamma}}$. Using this inequality in (3.5), it follows that for $k > 1 + \frac{r}{\gamma}$,

$$\mathbb{P}[E^{[k]} \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^*] > \prod_{i=(k-1)\gamma}^{k\gamma-1} i^{-\frac{1}{\gamma}} > ((k\gamma)^{-\frac{1}{\gamma}})^{\gamma} = \frac{1}{k\gamma}.$$

This finally provides the desired bound on the right-hand side of (3.3), so that

$$\mathbb{P}[\#A_{\mathbf{z}_{k\gamma}} < m^* \mid \#A_{\mathbf{z}_{(k-1)\gamma}} < m^*] \leq 1 - \frac{1}{k\gamma}. \quad (3.6)$$

Using (3.6), we now find

$$\mathbb{P}[\#A_{\mathbf{z}_{k\gamma}} < m^*] = \mathbb{P}[\#A_{\mathbf{z}_0} < m^*] \times \prod_{j=1}^k \mathbb{P}[\#A_{\mathbf{z}_{j\gamma}} < m^* \mid \#A_{\mathbf{z}_{(j-1)\gamma}} < m^*] \quad (3.7)$$

$$\leq \prod_{j=2+\lfloor \frac{r}{\gamma} \rfloor}^k \left(1 - \frac{1}{j\gamma}\right), \quad (3.8)$$

which settles the claim of part i).

ii) Since

$$\lim_{k \rightarrow \infty} \log \prod_{j=2+\lfloor \frac{r}{\gamma} \rfloor}^k \left(1 - \frac{1}{j\gamma}\right) \leq - \lim_{k \rightarrow \infty} \sum_{j=2+\lfloor \frac{r}{\gamma} \rfloor}^k \frac{1}{j\gamma} = -\infty,$$

the right-hand side of (3.8) converges to zero when $k \rightarrow \infty$, and since the events $\{\#A_{\mathbf{z}_i} < m^*\}$ are nested, we have

$$\mathbb{P}\left[\bigcap_{i \in \mathbb{N}} \{\#A_{\mathbf{z}_i} < m\}\right] = \lim_{k \rightarrow \infty} \mathbb{P}[\#A_{\mathbf{z}_{k\gamma}} < m] = 0.$$

□

We note that the proof of Theorem 2 can be modified to obtain similar convergence results for other variants of the RTR method applied to the MAXFS problem. In particular, for the singly RTR variant with $p_i = \exp(-v_i/t_i)$ and $\eta_i = t_i/t_0$ additional difficulties arise since p_i and η_i jointly decrease.

4 Some computational results

In our theoretical analysis, we have considered long non-overlapping sequences of updates and very slow temperature decreasing schedules. By weakening the mutual dependence of the events of finding an optimal solution during particular epochs, we made it possible to use arguments akin to the law of large numbers. It is however important to stress that our theoretical bounds on the temperature schedules are overly cautious in practice, and that our RTR methods perform very well with much faster linearly decreasing temperature schedules. We have devised a simple way to select the initial temperature based on the average inequality violation: t_0 is renormalized after each random cycle through all the inequalities as $t_0 := \frac{1}{3}t_0 + \frac{2}{3} \sum_k v_i^k$. We also use an adaptive scheme to decrease t_i to zero over a maximum number of cycles, max_cycles , which is effective on infeasible systems coming from a variety of applications: $t_i := \left(1 - \frac{cycle_index}{max_cycles}\right) t_0$.

To tackle large-scale instances of MAXFS, we have developed an efficient RTR implementation. Good sub-optimal solutions are obtained more rapidly by considering a block-iterative variant in which the update direction is given by a (convex) combination of the \mathbf{a}_k 's of the violated inequalities in a block and the size of the block is decreased over the iterations. See e.g. [6] for the block idea in the feasible case.

At each iteration we can look for a point of the segment $\{\alpha \mathbf{x}_i + (1 - \alpha) \mathbf{x}_{i+1}, \alpha \in [0, 1]\}$ connecting \mathbf{x}_i and \mathbf{x}_{i+1} that satisfies the largest number of inequalities. Despite the higher complexity (nm for the group subgradient version), this simple 1-D search yields moderate improvements.

After a certain number of non-improving iterations, a local search step is applied w.r.t. single variables. Given the current solution $\mathbf{x} = (x_1, x_2 \dots, x_n)$, we look for a variable index h , $1 \leq h \leq n$, and a scalar δ such that the number of inequalities satisfied by $(x_1, x_2 \dots, x_h + \delta \dots, x_n)$ is maximum. These local search steps, which have a complexity of nm , lead to better solutions. Encouraged by the performance of the RTR block version, we have adopted a grouping strategy also for this variable-based local search.

We now outline the results obtained for large-scale MAXFS instances arising from digital broadcasting [21] and from modeling protein folding potentials [16]. All inequalities that were not compatible with the variable upper and lower bounds have been eliminated in a preprocessing step. Moreover, additional orthogonal projections were used to take into account the upper and lower bounds on the variables.

Digital Video Broadcasting (DVB). In planning a DVB network, an interesting problem is that of determining the emission power of a given set of n transmitters so as to maximize territory coverage [21]. Suppose the territory is subdivided into q sufficiently small squared areas, considered as test points (TPs). A signal emitted from a transmitter is useful or interfering at a TP depending on the arrival delay. In the single frequency and fixed window model [21], for each TP i the signal quality constraint can be linearized as follows:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad (4.1)$$

where the variable x_j indicates the emission power of the j th transmitter, the field strength a_{ij} of the signal arriving at TP i from transmitter j is positive (negative) for the useful (interfering) part of the signal, and b_i is the minimum field strength required to cover TP i with probability 0.95. Since total coverage is usually not achievable, one faces sparse instances of MAXFS with one inequality per TP, a variable per transmitter and mandatory upper (maximum power) and

lower bounds for each variable. A reasonable discretization yields instances for the whole Italian territory with of the order of 55000 inequalities in a few thousands of variables. To maximize the population covered, a weight can be assigned to each inequality (4.1) and one searches a Feasible Subsystem (FS) of maximum total weight. See [21] for the modeling details.

Tests were carried out with a 2.8 GHz processor and 2 GB RAM on instances of different size m and n . In Table 1 the largest feasible subsystems obtained with RTR and the CPU time to find them are reported for a selection of instances. The results and performance are compared with Cplex 8.1 MIP solver applied to the big- M formulations. Although a reasonable value of M is easily derived from the variable bounds, it is quite large due to the problem nature and it makes the linear relaxation very poor. The time limit of 2 hours allowed Cplex to solve optimally only the smallest instances and one medium-size instance, and the best primal solutions (reported in the table with the time needed to find them) are often much worse than the approximate solutions provided by the RTR method. Moreover, RTR typically finds a good solution almost immediately and improves it subsequently. As an example, for instance *dvb5* RTR finds a FS of size 3346 after 0.16 sec. and improves it to 3354 within 2.33 sec., while Cplex finds a FS of size 3314 after 41.13 sec. and improves it to 3336 after 77089.65 sec. It is worth pointing that, because of the wide range of values the system coefficients can take, the best available (LP-based) heuristic [7] and the exact method presented at IPCO 2004 [8] failed out of numerical problems on all these instances.

			RTR		CPLEX	
	m	n	FS size	time in sec.	FS size	time in sec.
DIGITAL VIDEO BROADCASTING						
dvb1	550	487	538*	0.01	537 / 538*	0.02 / 0.36
dvb2	912	487	867 / 870*	0.01 / 19.58	850 / 870*	0.02 / 14.28
dvb3	1580	487	1531 / 1533	2.09 / 13.99	1515 / 1534*	2.00 / 24.38
dvb4	3767	487	3346 / 3354	0.16 / 2.33	3314 / 3336	41.13 / 77089.65
dvb6	6649	487	5757	46.88	–	–
dvb7	14057	487	12819	128.38	12365	58.20
dvb8	14605	487	9475	1.16	9480*	36.97
dvb9	19916	487	17154	167.49	15754	246.75
dvb10	8701	487	7479	12.09	–	–
dvb11	48758	487	41884	110.94	40082	1562.43
dvb12	54896	1681	42740	159.10	–	–
PROTEIN FOLDING POTENTIALS						
prot1	19404	12	16772	1.22	–	–
prot2	92890	212	92383	242.80	–	–
prot3	250233	150	250225	92.85	–	–

Table 1: Comparison between the RTR method and Cplex 8.1 MIP solver applied to big- M formulations for some DVB and Protein Folding instances; \star indicates proved optimal solutions and – no feasible solution.

Protein Folding. As described in [16], the problem of modeling the energy function (potential) underlying the folding of amino acid sequences into proteins gives rise to large and dense linear systems with millions up to tens of millions inequalities in hundreds of variables. Let $E(\mathbf{s}, \mathbf{t})$ denote the energy of sequence \mathbf{s} when folded into the three-dimensional structure \mathbf{t} and let \mathbf{t}_s^*

be the native structure of sequence \mathbf{s} . The premise that the energy of the native structure is lower than that of any structure chosen from a set of “decoys” amounts to

$$E(\mathbf{s}, \mathbf{t}) - E(\mathbf{s}, \mathbf{t}_s^*) > 0 \quad \forall \mathbf{s}, \forall \mathbf{t} \neq \mathbf{t}_s^*. \quad (4.2)$$

Focusing on the common features, potential modeling can then be formalized as the problem of approximating the unknown energy E by a linear combination of some appropriate basis function set $\{\phi_i(\mathbf{s}, \mathbf{t})\}_{1 \leq i \leq n}$, i.e., by $\tilde{E}(\mathbf{s}, \mathbf{t}) = \sum_{i=1}^n x_i \phi_i(\mathbf{s}, \mathbf{t})$ where \mathbf{x} is the vector of parameters. As a simple example, $\phi_i(\mathbf{s}, \mathbf{t})$ may count the number of contacts between a certain pair of amino acids that appear when \mathbf{s} is folded into structure \mathbf{t} . Thus, by requiring that the potential models \tilde{E} satisfy (4.2), we have $\sum_{i=1}^n x_i (\phi_i(\mathbf{s}, \mathbf{t}) - \phi_i(\mathbf{s}, \mathbf{t}_s^*)) > 0$ for all \mathbf{s} and all $\mathbf{t} \neq \mathbf{t}_s^*$. Since perfect structure recognition is unrealistic, these dense homogeneous systems are generally infeasible and one is interested in finding an \mathbf{x} satisfying as many inequalities (4.2) as possible [16].

As shown in Table 1, these instances are much larger than those from the DVB set, and the three corresponding big- M formulations could not be solved within the time limit of two hours and did not provide any primal solution. It is worth noting that for *prot3*, the FS of size 249,865 found by the RTR variant after 9 seconds is improved to 250,225, with only 8 violated inequalities, just after 92.85 seconds.

We have also tested the RTR method on several classification problems, see [7, 8]. For medium-to-large-size instances it tends to perform better than alternative approaches, such as the polyhedral methods [8, 19] and the LP-based heuristic [7]. For example, for the largest instance considered in [8] (*SolarFlare1066* with $m = 1066$) RTR found a FS of size 822 in less than 23 sec., while the best primal solutions reported in [8] and found in three hours are of size 793 for the big- M formulation and of size 782 for the algorithm based on combinatorial Benders’ cuts.

As our RTR methods do not require matrix inversions or computationally expensive matrix operations, they are particularly suited for large instances of MAXFS and very promising for huge ones. We are currently investigating ways to exploit the inherent parallelism to tackle the instances with tens of millions of inequalities arising in protein folding.

References

- [1] E. Amaldi and R. Hauser. Boundedness theorems for the relaxation method. Under minor revision for *Mathematics of Oper. Res.*, available from Optimization Online.
- [2] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147:181–210, 1995.
- [3] E. Amaldi, M. E. Pfetsch, and L. E. Trotter Jr. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs. *Math. Programming A*, 95:533–554, 2003.
- [4] K. P. Bennett and E. Bredensteiner. A parametric optimization method for machine learning. *INFORMS Journal on Computing*, 9:311–318, 1997.
- [5] H. D. Block and S. A. Levin. On the boundedness of an iterative procedure for solving a system of linear inequalities. In *Proceedings of AMS*, pages 229–235, 1970.
- [6] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, algorithms and applications*. Oxford University Press, 1997.

- [7] J. Chinneck. Fast heuristics for the maximum feasible subsystem problem. *INFORMS Journal on Computing*, 13:210–213, 2001.
- [8] G. Codato and M. Fischetti. Combinatorial Benders’ cuts. In *Proceedings of IPCO*, volume 3064 of *LNCS*, pages 178–195. Springer, 2004.
- [9] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In *Proceedings of STOC*, pages 315–320. ACM Press, 2004.
- [10] M. Frean. A “thermal” perceptron learning rule. *Neural Comp.*, 4(6):946–957, 1992.
- [11] J. L. Goffin. The relaxation method for solving systems of linear inequalities. *Mathematics of Oper. Res.*, 5:388–414, 1980.
- [12] H. J. Greenberg and F. H. Murphy. Approaches to diagnosing infeasible linear programs. *ORSA Journal on Computing*, 3:253–261, 1991.
- [13] E. K. Lee, R. J. Gallagher, and M. Zaider. Planning implants of radionuclides for the treatment of prostate cancer: An application of MIP. *Optima*, 61:1–7, 1999.
- [14] O. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer et al., editor, *Applied Mathematics and Parallel Computing*, pages 175 – 188. Physica-Verlag, 1996.
- [15] M. Mattavelli, V. Noel, and E. Amaldi. Fast line detection algorithms based on combinatorial optimization. In *Proceedings of IWVF*, LCNS 2059, pages 410–419. Springer, 2001.
- [16] J. Meller, M. Wagner, and R. Elber. Solving huge linear programming problems for the design of protein folding potentials. *Math. Programming B*, 101:301–318, 2004.
- [17] M. L. Minsky and S. Papert. *Perceptrons: An introduction to computational Geometry*. MIT Press, Cambridge, MA, 1988. Expanded edition.
- [18] A. Nedić and D. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 12:109–138, 2001.
- [19] M. E. Pfetsch. *The maximum feasible subsystem problem and vertex-facet incidences of polyhedra*. PhD thesis, Dep. of Mathematics, Technische Universität Berlin, October 2002.
- [20] B. T. Polyak. Random algorithms for solving convex inequalities. In D. Butnariu et al., editor, *Inherently parallel algorithms in feasibility and other applications*. Elsevier, 2001.
- [21] F. Rossi, A. Sassano, and S. Smriglio. Models and algorithms for terrestrial digital broadcasting. *Ann. of Oper. Res.*, (107):267–283, 2001.
- [22] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley & Sons, 1986.

Acknowledgments

The authors are grateful to John Chinneck (Gianni Codato and Matteo Fischetti) for experimenting with his (their) LP-based heuristic (polyhedral approach) on the DVB data. They also would like to thank Fabrizio Rossi and Stefano Smriglio (Michael Wagner and Jarek Meller) for providing the DVB (protein folding) data. Raphael Hauser was supported through grant NAL/00720/G from the Nuffield Foundation and through grant GR/M30975 from the Engineering and Physical Sciences Research Council of the UK.