A first-order ϵ-approximation algorithm for large linear programs

João Soares*

December 19, 2000

Abstract

This report presents an algorithm that finds an ϵ-feasible solution relatively to some constraints of a linear program. The algorithm is a first-order feasible directions method with constant stepsize that attempts to find the minimizer of an exponential penalty function. When embedded with bisection search, the algorithm allows for the approximated solution of linear programs. The running time of our algorithm depends polynomially on $1/\epsilon$ and a parameter *width* introduced by Plotkin, Shmoys and Tardos in [3] and it is especially interesting when the direction finding (linear) subproblem is considered *easy* and amenable to reoptimization. We present applications of this framework to the Held and Karp bound on the traveling salesman problem and to a class of hard 0-1 linear programs. Computational results are expected to complement this report in the forthcoming revised version.

1 Introduction

Some linear programs arising from real-world applications have such a large number of variables and/or constraints that they can't be dealt with simplex-type methods, or even interior-point methods. This is the case with most of the fractional set-covering, -partitioning and -packing models, that usually have a abnormally large number of variables. Models like these arise in applications like crew scheduling (trains, buses or airplanes), political districting, protection of microdata, information retrieval, etc. Tipically these models are suboptimally solved by heuristics because an optimization framework (usually of the branch-and-price type) has to be rather specialized, if feasible at all. Moreover, a branch-and-price framework requires the solution of large linear programs at every node of the branch-and-price tree and these linear programs may take a long time and storage to be solved to optimality.

Our framework attempts to find reasonable approximate solutions to those models quickly and without too much storage, along the lines of Lagrangian relaxation. The approximation obtained may serve the purpose of speeding-up the optimal basis identification. We will be

*Mathematics Department, University of Coimbra, 3000 Coimbra, Portugal. Email: `jsoares@mat.uc.pt`

looking for an approximated solution of a linear program in the following form

$$
\begin{aligned}
z^* \equiv \min \quad & cx \\
\text{s.t.} \quad & Ax \geq b, \\
& x \in P,
\end{aligned}
\tag{1}
$$

where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $P \subseteq \mathbb{R}^n$ is a set (possibly, a lattice) over which optimizing linear programs is considered "easy". For example, in a set covering model the matrix A is a matrix of zeros and ones, the vector b is a vector of all-ones, and the set P is the lattice $\{0,1\}^n$, or the hypercube $[0,1]^n$ in the fractional version. If P includes a budget constraint then P becomes the feasible region of a knapsack problem or a continuous knapsack problem, respectively.

We focus on obtaining a reasonable approximation to the optimal solution of (1) by an ϵ-feasible solution. A point $x \in P$ is ϵ-feasible relatively to the constraints $Ax \geq b$ if

$$\lambda(x) \equiv \max_{i=1,\ldots,m} b_i - a_i x \leq \epsilon \quad \Longleftrightarrow \quad b - Ax \leq \epsilon e,$$

where a_i denotes the ith row of the matrix A and e denotes a column vector of all-ones. To achieve this, we propose a first-order feasible directions method with constant stepsize that attempts to solve the nonlinear program

$$
\begin{aligned}
\Phi(\alpha, z) = \min \quad & \phi(x) \equiv \sum_{i=0}^{m} \exp(\alpha\,(b_i - a_i x)) \\
\text{s.t.} \quad & x \in P' \equiv P \cap \{x \colon cx \leq z\}
\end{aligned}
\tag{2}
$$

for given values of the parameters α and z, where $\exp(\cdot)$ denotes the exponential function. The scalar α is a penalty parameter and z is a guess for the value of z^*. The running time of our algorithm depends polynomially on $1/\epsilon$ and the *width* of the set P' relatively to the constraints $Ax \geq b$. Significantly, the running time does not depend explicitly on n and, hence, it can be applied when n is exponentially large, assuming that, for a given row vector y, there exists a polynomial subroutine to optimize yAx over P'.

All the problems in our framework are known to be solvable in polynomial time, but the amount of computer storage can be so high that the complexity may not be observable in practice. Thus, our interest is in proposing an algorithm that obtains an ϵ-approximation to the optimal solution through solving a sequence of linear programs that are *easy* while, for a fixed ϵ, guaranteeing a pseudo-polynomial complexity in the worst case.

2 The algorithm

We assume that the feasible region of program (2) is nonempty and bounded for any z of interest so that, at least, an optimal solution exists. We say that $\bar{x}$ is ϵ-optimal if $\bar{x} \in P'$ and $\phi(\bar{x}) \leq (1+\epsilon)\Phi(\alpha, z)$.

If x is feasible in (1) then $\phi(x) \leq m$, while if x is simply ϵ-feasible then $\phi(x) \leq m \exp(\alpha\epsilon)$. On the other hand, if x is not ϵ-feasible then $\exp(\alpha\epsilon) < \phi(x)$. Clearly, if $\bar{x}$ is ϵ-optimal with $\phi(\bar{x}) > (1+\epsilon)m$ then there is no feasible solution x in (1) with $cx \leq z$. We will choose α so that it may be possible to assert whether $\bar{x}$ is ϵ-feasible from the value of $\phi(\bar{x})$, as formally stated in the next proposition.

Lemma 1 *Assume that $\alpha \geq \ln((1+\epsilon)m)/\epsilon$. Then,*

1. *If there is no ϵ-feasible solution in (1) such that $cx \leq z$ then $\Phi(\alpha, z) > (1+\epsilon)m$.*
2. *If $\bar{x}$ is feasible in (2) and $\phi(\bar{x}) \leq (1+\epsilon)m$ then $\bar{x}$ is ϵ-feasible with $c\bar{x} \leq z$.*

Proof: Use $\epsilon < \epsilon' \equiv \min\{\lambda(x) : x \in P, cx \leq z\}$ to prove the first assertion. If $\phi(\bar{x}) \leq (1+\epsilon)m$ then, for any $i = 0, \ldots, m$, $\exp(\alpha\,(b_i - a_i\bar{x})) \leq (1+\epsilon)m$, or equivalently, $b_i - a_i\bar{x} \leq \ln((1+\epsilon)m)/\alpha \leq \epsilon$. □

Thus, keeping the value of α fixed, we may use bisection to search for the minimum value of z that produces a feasible solution x in (1) with $cx \leq z$. The bisection search maintains an interval $[z_a, z_b]$ such that there is no feasible solution x in (1) with $cx \leq z_a$ and there is a known ϵ-feasible solution x in (1) with $cx \leq z_b$. The search is interrupted when $z_b - z_a \leq \epsilon z_b$. This does not imply any bound on how much z_b differs from z_*. It may be possible that z_b is much less than z_* though very unlikely for α large, as our next result shows.

Proposition 1 *Let the sequence $\{\alpha_k\}$ be such that $\lim_k \alpha_k = +\infty$ and $y^k = [\alpha_k \exp(\alpha_k(b_i - a_i x^k))]_i$, where x^k is optimal in $\Phi(\alpha_k, z_*)$. Then, every accumulation point $(\bar{x}, \bar{y})$ of the sequence $\{(x^k, y^k)\}$ is such that $\bar{y} \geq 0$, $A\bar{x} \leq b$ and*

$$(c - \bar{y}A)\,(x - \bar{x}) \geq 0, \tag{3}$$

for every $x \in P$, i.e., $\bar{x}$ is optimal for program (1).

Proof: Assume $\lim_{k \in K}(x^k, y^k) = (\bar{x}, \bar{y})$. By continuity, we have that $\bar{x} \in P, \bar{y} \geq 0$ and $(-\bar{y}A)\,(x - \bar{x}) \geq 0$, for every $x \in P \cap \{x : cx \leq z_*\}$. From the definition of $\bar{y}_i$ we conclude that $a_i\bar{x} \leq b_i$, for every i, and that $\bar{y}_i$ must be zero whenever $a_i\bar{x} < b_i$. Thus, $A\bar{x} \leq b$, which implies that $\bar{x}$ is optimum for program (1), and $\bar{y}(A\bar{x} - b) = 0$. From this it is easy to conclude that (3) holds for any x that is optimal for program (1). When x be feasible for program (1) but nonoptimal, we have that $\bar{y}A(x - \bar{x}) \leq 0$ and $c(x - \bar{x}) > 0$ so that $(c - \bar{y}A)(x - \bar{x}) > 0$, also. □

If we assume a unique correspondence between α and ϵ then Proposition 1 induces the following practical modification to the bisection search described before. Let $[z_a^k, z_b^k]$ be the last bisection interval of the search for a given ϵ_k, thus, satisfying $z_b^k - z_a^k \leq \epsilon_k z_b^k$. Then, restart the bisection search with $[z_b^k, z_b]$ for some $\epsilon_{k+1} < \epsilon_k$, for example, $\epsilon_{k+1} = \epsilon_k/2$. Note that ϵ_k-feasible solutions are known at both extremes of the new initial interval (and possibly in many others in between) and, thus, the initial solutions for the minimization of ϕ when $\epsilon = \epsilon_{k+1}$ will not *overflow* the exponential functions evaluation by too much. From Proposition 1, z_b^k goes to z_* as ϵ_k goes to zero.

Our algorithm for solving (2) is a first-order iterative procedure. The algorithm coincides with the algorithm Improve-Packing proposed in [3] but the stepsize and the stopping criterion are different. Starting from $x^0 \in P'$, at a generic iterate $\bar{x}$ that is not ϵ-feasible the direction of movement is determined from solving the linear program

$$\begin{array}{ll} \min & m_1(x) \equiv \phi(\bar{x}) + \nabla\phi(\bar{x})(x - \bar{x}) \\ \text{s.t.} & x \in P' = P \cap \{x : cx \leq z\} \end{array} \tag{4}$$

Then, we reset $\bar{x}$ to $\bar{x} + \hat{\sigma}(\hat{x} - \bar{x})$, for some fixed stepsize $\hat{\sigma} \in (0, 1]$, and proceed analogously to the next iteration. The algorithm is halted when $\phi(\bar{x}) \leq m(1 + \epsilon)$ or a maximum number of iterations is reached.

Theorem 1 below presents one particular choice for the stepsize that depends on the following quantity, introduced as the *width* of P relatively to the constraints $Ax \geq b$ in [3],

$$\rho \equiv \max_{i=1,2,\ldots,m} \left(\begin{array}{ll} \max & \mid a_i x - a_i y \mid \\ \text{s.t.} & x, y \in \text{conv (ext } P') \end{array} \right), \tag{5}$$

where conv $(\cdot)$ denotes convex hull and ext P' denotes the set of extreme points of conv (P'). Hence, conv (ext P') is always a polytope and ρ is well defined. In [3] ρ is defined differently depending when whether the matrix A is such that $Ax \geq 0$, for every $x \in P$, or not. If yes then the two definitions coincide with $\rho = \max_i \max_{x \in P} a_i x$. If not then our definition of ρ is half of the ρ that is proposed in [3]. We also recall that the stepsize proposed in [3], which is $\epsilon/(4\rho\alpha)$, is much smaller than the stepsize that we propose in the following theorem.

Theorem 1 *Assume that* $z_* \leq z$, $\bar{x} \in$ conv (ext P') *is not* ϵ*-feasible,* $\epsilon \in (0, 1)$, $\hat{x} \in$ ext P' *is optimal for program (4),* ρ *is given by (5) and*

$$\alpha \geq \max\left(\frac{\ln(m(3 + \epsilon))}{\epsilon}, \frac{1}{\rho \ln 2}\right). \tag{6}$$

Then, for $\hat{\sigma} = 1/(\alpha\rho)^2$ *we have that*

$$\phi\left(\bar{x} + \hat{\sigma}(\hat{x} - \bar{x})\right) < \left(1 - \frac{1}{4(\alpha\rho)^2} \frac{1 + \epsilon}{3 + \epsilon}\right) \phi(\bar{x}).$$

Proof: Let $g(\sigma) \equiv \phi\left(\bar{x} + \sigma(\hat{x} - \bar{x})\right)$, for $\sigma \in (0, 1]$. From the second-order Taylor expansion of g, there exists $\xi \in (0, \sigma]$ such that

$$g(\sigma) = g(0) + \sigma g'(0) + \frac{\sigma^2}{2} g''(\xi), \tag{7}$$

where $g(0) = \phi(\bar{x})$, $g'(0) = \nabla\phi(\bar{x})^T(\hat{x} - \bar{x})$ and $\sigma^2 g''(\xi)/2 = \sum_{i=1}^m \left(\sigma\alpha a_i \left(\hat{x} - \bar{x}\right)\right)^2 \exp(r_i)$, where r_i lies between $\alpha(b_i - a_i\bar{x})$ and $\alpha(b_i - a_i\bar{x}) + \sigma\alpha a_i(\hat{x} - \bar{x})$. If $\mid \sigma\alpha a_i(\hat{x} - \bar{x}) \mid \leq \delta$ then, from the second-order Taylor expansion of the exponential,

$$\exp(r_i) \leq \exp(\alpha(b_i - a_i\bar{x})) \left[1 + \alpha\sigma a_i(\hat{x} - \bar{x}) + \frac{\delta^2 \exp(\delta)}{2} + \delta\right]$$

which implies that

$$\begin{aligned}\frac{\sigma^2}{2}g''(\xi) &\le \frac{\delta^2}{2}\sum_{i=1}^{m}\exp(\alpha(b_i - a_i\bar{x}))\left[1+\alpha\sigma a_i(\hat{x}-\bar{x}) + \frac{\delta^2\exp(\delta)}{2} + \delta\right] \\ &= \frac{\delta^2}{2}\left(1+\frac{\delta^2\exp(\delta)}{2}+\delta\right)\phi(\bar{x}) + \sigma\frac{\delta^2}{2}\nabla\phi(\bar{x})(\hat{x}-\bar{x})\end{aligned}$$

Thus, from (7),

$$g(\sigma) \le \left[1+\frac{\delta^2}{2}\left(1+\frac{\delta^2\exp(\delta)}{2}+\delta\right)\right]\phi(\bar{x}) + \sigma\left(1+\frac{\delta^2}{2}\right)\nabla\phi(\bar{x})(\hat{x}-\bar{x}). \tag{8}$$

Since $z_* \le z$, there exists x^* feasible in (1) that is also feasible in (4). Then, using the fact that $\bar{x}$ is not ϵ-feasible,

$$\phi(x^*) \le \frac{m}{\exp(\alpha\epsilon)}\exp(\alpha\epsilon) \le \frac{m}{\exp(\alpha\epsilon)}\phi(\bar{x}) \le \frac{1}{3+\epsilon}\phi(\bar{x}).$$

Now, from the convexity of ϕ, we have that

$$\nabla\phi(\bar{x})(\hat{x}-\bar{x}) \le \nabla\phi(\bar{x})(x^*-\bar{x}) \le -\phi(\bar{x}) + \phi(x^*) \le -\frac{2+\epsilon}{3+\epsilon}\phi(\bar{x}). \tag{9}$$

By plugging (9) into (8), we obtain

$$g(\sigma) \le \left[1+\frac{\delta^2}{2}\left(1+\frac{\delta^2\exp(\delta)}{2}+\delta\right) - \sigma\left(1+\frac{\delta^2}{2}\right)\left(\frac{2+\epsilon}{3+\epsilon}\right)\right]\phi(\bar{x}).$$

Now, if we choose σ to satisfy $\alpha\sigma\rho \le \delta$ (i.e., $\sigma = \delta/(\alpha\rho)$) then $g(\sigma) \le \Delta(\delta)\phi(\bar{x})$, where

$$\Delta(\sigma) \equiv 1+\frac{\delta^2}{2}\left(1+\frac{\delta^2\exp(\delta)}{2}+\delta\right) - \frac{\delta}{\alpha\rho}\left(1+\frac{\delta^2}{2}\right)\frac{2+\epsilon}{3+\epsilon}$$

We need to choose δ such that $\Delta(\delta) < 1$ and as small as possible. If we restrict our attention to δ such that $1 < \exp(\delta) \le 2$ (i.e., $0 < \delta \le \ln 2$) then

$$\begin{aligned}\Delta(\delta) &\le 1+\frac{\delta^2}{2}\left(1+\delta^2+\delta\right) - \frac{\delta}{\alpha\rho}\left(1+\frac{\delta^2}{2}\right)\frac{2+\epsilon}{3+\epsilon} \\ &= 1+\frac{\delta^2}{2}\left(1+\delta^2\right) - \frac{\delta}{\alpha\rho}\left(1+\delta^2\right)\frac{2+\epsilon}{3+\epsilon} + \frac{\delta^3}{2\alpha\rho}\frac{2+\epsilon}{3+\epsilon} + \frac{\delta^3}{2}.\end{aligned}$$

At $\hat{\delta} = 1/(\alpha\rho)$ we have that

$$\begin{aligned}\Delta(\hat{\delta}) &\le 1+\frac{1}{(\alpha\rho)^2}\left(1+\frac{1}{(\alpha\rho)^2}\right)\left(\frac{1}{2}-\frac{2+\epsilon}{3+\epsilon}\right) + \frac{1}{2(\alpha\rho)^4}\frac{2+\epsilon}{3+\epsilon} + \frac{1}{2(\alpha\rho)^3} \\ &= 1-\frac{1}{2(\alpha\rho)^2}\frac{1+\epsilon}{3+\epsilon} + \frac{1}{2(\alpha\rho)^3} + \frac{1}{2(\alpha\rho)^4}\frac{2+\epsilon}{3+\epsilon} \quad < 1-\frac{1}{4(\alpha\rho)^2}\frac{1+\epsilon}{3+\epsilon}.\end{aligned}$$

□

In summary, assuming that $z_* \leq z$, if the first k iterates are not ϵ-feasible then

$$\phi(x^{k+1}) < \left(1 - \frac{1}{4(\alpha\rho)^2}\frac{1+\epsilon}{3+\epsilon}\right)^k \phi(x^0),$$

where we note that the right hand side goes to zero. The following corollary states a worst-case complexity bound on the number of iterations of the algorithm.

Corolary 1 *If α satisfies (6) and $\epsilon \in (0,1)$ then our algorithm, using $\hat{\sigma} = 1/(\alpha\rho)^2$ and starting from $x^0 \in P$ such that $cx \leq z$, terminates after*

$$\frac{\ln(m) + \ln(1+\epsilon) - \ln\phi(x^0)}{\ln\left(1 - \frac{1}{4(\alpha\rho)^2}\frac{1+\epsilon}{3+\epsilon}\right)} < 16\alpha^3\rho^2\lambda(x^0) \tag{10}$$

iterations, with an ϵ-feasible solution or, otherwise, with the proof that no feasible solution for program (1) such that $cx \leq z$ exists.

Proof: If $z^* \leq z$ then the left-hand-side of (10) is an upper bound on the number of iterations k such that $\phi(x^{k+1}) < m(1+\epsilon)$. When this is the case then, from Lemma 1, x^{k+1} is ϵ-feasible. If, still after that many iterations, no ϵ-feasible solution was found then, from Theorem 1, $z < z_*$. By using the fact that $-1/\ln(1-x) < 1/x$, for any $x \in (0,1)$, we have that

$$\frac{\ln(m) + \ln(1+\epsilon) - \ln\phi(x^0)}{\ln\left(1 - \frac{1}{4(\alpha\rho)^2}\frac{1+\epsilon}{3+\epsilon}\right)} < \Big(\alpha\lambda(x^0) - \ln(m) - \ln(1+\epsilon)\Big)\, 4(\alpha\rho)^2\frac{3+\epsilon}{1+\epsilon},$$

from where (10) easily follows. □

We remark that the right hand side of (10) is $\mathcal{O}(\ln^3(m)\epsilon^{-3}\rho^2\lambda(x^0))$. If $\lambda(x^0) = \mathcal{O}(\epsilon)$ then only $\mathcal{O}\left(\ln^3(m)\epsilon^{-2}\rho^2\right) = \tilde{\mathcal{O}}\left(\epsilon^{-2}\rho^2\right)$ iterations are required. This complexity result is related to Karger and Plotkin's [2, Theorem 2.5] ($\tilde{\mathcal{O}}\left(\epsilon^{-3}\rho\right)$) and Plotkin, Shmoys and Tardos' [3, Theorem 2.12] ($\tilde{\mathcal{O}}(\epsilon^{-2}\rho\ln\epsilon^{-1})$). The result of Karger and Plotkin is valid even if the budget constraint is included in the objective function of (2) without counting in the definition of ρ. We recall that a function $f(n)$ is said to be $\tilde{\mathcal{O}}(g(n))$ if there exists a constant c such that $f(n)\ln^c(n) \geq \mathcal{O}(g(n))$.

3 Applications

In this section, we show how to apply the algorithm presented in the previous section to find an ϵ-approximation to a variety of optimization problems.

3.1 Held and Karp lower bound on the TSP

The Held and Karp lower bound is a lower bound on the value of a minimum length traveling salesman tour on an undirected graph $G = (V, E)$, where V denotes the node set and E

denotes the edge set. We assume an instance of costs c_e, for $e \in E$, that are positive and satisfy the triangle inequality.

Held and Karp [?] showed that the lower bound, denoted z_*, can be obtained through a subgradient algorithm. They have also showed that z_* is the optimal value over the *subtour elimination polytope*, i.e., following closely the explanation given in [3],

$$\begin{aligned} z_* = \min \quad & \sum_{e \in E} c_e y_e \\ \text{s.t.} \quad & \sum_{e \in E \cap S_i} y_e \geq 2, \quad i = 1, 2, \ldots, s \\ & y_e \geq 0, \quad e \in E \end{aligned}$$

where $\{S_i, i = 1, 2, \ldots, s\}$ denotes the family of all the cuts on the graph G. In this formulation, variables correspond to edges and there is a constraint for each cut. We shall instead focus on the dual of this linear program:

$$\begin{aligned} z_* = \max \quad & 2 \sum_{j=1}^{s} x_j \\ \text{s.t.} \quad & \sum_{j=1}^{s} \left(\frac{\delta_{ej}}{c_e} \right) x_j \leq 1, \quad e \in E \\ & x_j \geq 0, \quad j = 1, 2, \ldots, s \end{aligned}$$

where $\delta_{ej} = 1$ if cut S_j separates the two ends of the edge e. In this formulation, variables correspond to cuts and there is a constraint for each edge.

Let $P' = \{x \colon \sum_{j=1}^{s} x_j \geq z/2, x \geq 0\}$ and the system $Ax \geq b$ be given by the remaining constraints. Since $b = e$, $x \in P$ is ϵ-feasible if $Ax \leq (1+\epsilon)b$. It is well known that, due to the triangle inequality, $z_* \in [C_{1T}, 2C_{1T}]$, where C_{1T} is the weight of a minimum weight 1-tree on the graph G then the *width* for this formulation satisfies

$$\rho = \frac{C_{1T}}{\min_{e \in E} c_e} \geq \frac{z}{2 \min_{e \in E} c_e} = \max_{e \in E} \left(\begin{aligned} \max \quad & \sum_{j=1}^{s} \left(\frac{\delta_{ej}}{c_e} \right) x_j \\ \text{s.t.} \quad & \sum_{j=1}^{s} x_j = \frac{z}{2} \\ & x_j \geq 0, \quad j = 1, 2, \ldots, s \end{aligned} \right).$$

Construct an initial point $x^0 \in P'$ for the optimization of (2) in the following way: $(z/2)$ at the component correspondent to some cut S on the graph G and 0 anywhere else. The cut S separates the set V in two disjoint sets of vertices V_1 and V_2 and so, $\lambda(x^0) = z/(2 \min_{e \in \delta(V_1)} c_e) \leq \rho$. From Corollary 1, if we use $\alpha = \ln(|E|(3+\epsilon))/\epsilon$ then the number of iterations required by our algorithm to find an ϵ-feasible solution or to show that no feasible solution exists is $\mathcal{O}(\ln^3 \mid E \mid \epsilon^{-3} \rho^3)$. The subprograms that are required to be solved at each iteration of our algorithm are min-cut problems and can be solved in $\mathcal{O}(\mid V \mid^3)$.

The bisection search can be initialized with the interval $[z_a, z_b] = [C_{1T}, 2C_{1T}]$ and ends when the amplitude of the interval $[z_a, z_b]$ is such that $z_b - z_a \leq \epsilon z_a$. At this point, it is known a solution $\bar{x} \in P'$, in particular $\sum_j \bar{x}_j \geq (z_a/2)$, with $A\bar{x} \leq (1+\epsilon)b$, and no feasible $x \in P$

exists, in particular $\sum_j x_j \geq (z_b/2)$, such that $Ax \leq b$. The latter part of this implies that the Held and Karp bound is at most $z_a(1+\epsilon)$. The former part implies that $\bar{x}/(1+\epsilon)$ is a feasible solution to of value at least $z_a/(1+\epsilon)$. Therefore, $z_a/(1+\epsilon) \leq z_* \leq z_a(1+\epsilon)$, which implies that $z_a/(1+\epsilon)$ is a valid lower bound on the TSP that is within a factor of $(1+\epsilon)^2$ of the Held and Karp bound. The complexity of the overall algorithm in terms of calls to the linear subprograms is $\mathcal{O}(\ln^3 \mid E \mid \epsilon^{-3} \ln(1/\epsilon)\rho^3)$.

3.2 A class of hard small 0-1 programs

Cornuéjols and Dawandee proposed at the IPCO VI conference, later published in [1], a set of 0-1 linear programming instances that proved to be very hard to solve by traditional methods, and in particular by linear programming based branch-and-bound. This class of problems can be summarised as follows: Find a point $x \in \{0,1\}^n$ in the intersection of m hyperplanes in $\mathbb{R}^n$ that minimizes a linear function cx. Mathematically,

$$
\begin{aligned}
z_* = \min \quad & \sum_{j=1}^{n} c_j x_j & \qquad = \min \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} d_{ij} x_j = p_i, i = 1, \ldots, m & \qquad \text{s.t.} \quad & \sum_{j=1}^{s} \left(\frac{d_{ij}}{p_i}\right) x_j = 1, i = 1, \ldots, m \\
& x_j \in \{0,1\}, \quad j = 1, \ldots, n, & & x_j \in \{0,1\}, \quad j = 1, \ldots, n,
\end{aligned}
$$

where $D = [d_{ij}] \geq 0$, $p = [p_i] > 0$, $c = [c_j] > 0$ and $d_{ij} < p_i$, for every i, j. All scalars are integer numbers. We will assume, without loss of generality, that p is a vector of all-ones and that D is a matrix of real nonnegative numbers satisfying $d_{ij} < 1$, for any i, j.

Let $P' = \{x \in \{0,1\}^n \colon \sum_{j=1}^{n} c_j x_j \leq z\}$ and the system $Ax \geq b$, where $A \in \mathbb{R}^{2m \times n}$ and $b \in \mathbb{R}^{2m}$, be given by the remaining constraints $Dx \geq e$ and $-Dx \geq -e$. Thus, $x \in P'$ is ϵ-feasible if $(1-\epsilon)e \leq Dx \leq (1+\epsilon)e$. Since $Dx \geq 0$, for any $x \in P'$, the *width* for this formulation satisfies

$$
\rho = n \geq \max_{i=1,2,\ldots m} \left(\begin{array}{ll} \max & \sum_{j=1}^{n} d_{ij} x_j \\ \text{s.t.} & \sum_{j=1}^{n} c_j x_j \leq z \\ & x_j \in \{0,1\}, \quad j = 1, 2, \ldots, n \end{array} \right). \tag{11}
$$

Note that, if d_i denotes the ith row of the matrix D then the objective function in (2) is given by

$$
\phi(x) \equiv \sum_{i=1}^{m} \exp(\alpha(1 - d_i x)) + \exp(\alpha(d_i x - 1)).
$$

Let $x^0 \in P'$ be any starting point for the optimization of (2). Since $\lambda(x^0) \leq n$, if we use $\alpha = \ln(2m(3+\epsilon))/\epsilon$ then the number of iterations required by our algorithm to find an ϵ-feasible solution or to show that no feasible solution exists is $\mathcal{O}(\ln^3(2m)\epsilon^{-3}n^3)$. The subprograms that are required to be solved at each iteration of our algorithm are knapsack problems and can be solved in $\mathcal{O}(nz)$ operations by a dynamic programming algorithm.

The bisection search can be initialized with the interval $[z_a, z_b] = [\underline{z}, \bar{z}]$, where $\bar{z}$ is the optimal value of the linear programming relaxation of (11) and $\bar{z} \leq \sum_{j=1}^{n} c_j$ is a known upper bound on the value of z_*. The bisection search ends when the interval $[z_a, z_b]$ is such that $z_b - z_a \leq \epsilon z_b$. At this point, it is known a solution $\bar{x} \in P$ such that $(1-\epsilon)e \leq D\bar{x} \leq (1+\epsilon)b$ and $\sum_j c_j \bar{x}_j \leq z_b$, and no $x \in P$ satisfying $Ax = b$ exists such that $\sum_j c_j x_j \leq z_a$.

This report will be complemented with computational results on the Cornuéjols-Dawandee instances.

References

[1] G. Cornuéjols and M. Dawandee, *A class of hard small 0-1 programs*, INFORMS Journal on Computing **11** (1999), 205–210.

[2] D. Karger and S. Plotkin, *Adding multiple cost constraints to combinatorial optimization problems, with applications to multicommodity flows*, Proceedings of the 27th Annual ACM Symposium on Theory of of Computing, 1995.

[3] S. Plotkin, D. Shmoys, and E. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Mathematics of Operations Research **20** (1995), 257–301.