A Appendix

A.1 BLCP to LCP Reformulation

Here we generalize an observation from Júdice and Pires [19] to show that any BLCP can be converted to an LCP of *twice* the original dimension. Recall that the BLCP seeks vectors w and z such that $w = q + Mz$ and each variable z_i meets one of the following conditions:

$$\begin{aligned} z_i = u_i \quad &\Longrightarrow \quad w_i \leq 0 \\ z_i = l_i \quad &\Longrightarrow \quad w_i \geq 0 \\ l_i < z_i < u_i \quad &\Longrightarrow \quad w_i = 0 \end{aligned}$$

where the vectors l and u are lower and upper bounds respectively. An LCP must have nonnegative variables, so the BLCP conversion begins by introducing a nonnegative slack variable $z^+ = z - l$. This slight change allows for writing the problem as $w = (q + Ml) + Mz^+$ where:

$$\begin{aligned} z_i^+ = u_i - l_i \quad &\Longrightarrow \quad w_i \leq 0 \\ z_i^+ = 0 \quad &\Longrightarrow \quad w_i \geq 0 \\ 0 < z_i^+ < u_i - l_i \quad &\Longrightarrow \quad w_i = 0. \end{aligned}$$

The next step is to deal with the fact that variable w can be negative. We do so by introducing two sets of nonnegative variables such that $w = w^+ - w^-$. Now, the problem can be written $w^+ = (q + Ml) + Mz^+ + w^-$ where:

$$\begin{aligned} z_i^+ = u_i - l_i \quad &\Longrightarrow \quad w_i^+ = 0 \;\; \& \;\; w_i^- \geq 0 \\ z_i^+ = 0 \quad &\Longrightarrow \quad w_i^+ \geq 0 \;\; \& \;\; w_i^- = 0 \\ 0 < z_i^+ < u_i - l_i \quad &\Longrightarrow \quad w_i^+ = w_i^- = 0. \end{aligned}$$

At this point, the conditions can be expanded into an LCP with twice the original dimension. To help in this conversion, we add another slack variable $z^- = u - z = u - l - z^+$. Now the problem becomes:

$$\begin{bmatrix} w^+ \\ z^- \end{bmatrix} = \begin{bmatrix} q + Ml \\ u - l \end{bmatrix} + \begin{bmatrix} M & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} z^+ \\ w^- \end{bmatrix}$$

with variables $w^+, w^-, z^+, z^- \geq 0$ and $z_i^+ w_i^+ = z_i^- w_i^- = 0, \forall i$. If matrix M is invertible then we can pivot on M, yielding the following equivalent LCP:

$$\begin{bmatrix} z^+ \\ z^- \end{bmatrix} = \begin{bmatrix} -l - M^{-1}q \\ u + M^{-1}q \end{bmatrix} + \begin{bmatrix} M^{-1} & -M^{-1} \\ -M^{-1} & M^{-1} \end{bmatrix} \begin{bmatrix} w^+ \\ w^- \end{bmatrix}$$

with variables $w^+, w^-, z^+, z^- \geq 0$ and $z_i^+ w_i^+ = z_i^- w_i^- = 0, \forall i$. Given the solution to this LCP, the original solution to the BLCP can be computed via $w = w^+ - w^-$ and $z = z^+ + l$.

A.2 LCP formulation of the L_1 Regularized Linear Fixed Point

Here, we derive an LCP formulation of the L_1 reguarized fixed point directly. Some readers may find this presentation more intuitive than the BLCP formulation.

Begin with the LARS-TD optimality conditions. Each coefficient w_i must meet one of the following conditions:

$$\begin{aligned} -\beta < \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) < \beta \quad &\Longrightarrow \quad w_i = 0 \\ \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) = \beta \quad &\Longrightarrow \quad w_i \geq 0 \\ \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) = -\beta \quad &\Longrightarrow \quad w_i \leq 0 \end{aligned}$$

These can be rewritten as:

$$\begin{aligned} 0 < \beta - \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) < 2\beta \quad &\Longrightarrow \quad w_i = 0 \\ \beta - \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) = 0 \quad &\Longrightarrow \quad w_i \geq 0 \\ \beta - \Phi_i^T(R - (\Phi - \gamma\Phi'^{\pi})w) = 2\beta \quad &\Longrightarrow \quad w_i \leq 0 \end{aligned}$$

Now define two vectors x^+ and x^- as:

$$x^+ = \beta - \Phi^T(R - (\Phi - \gamma\Phi'^\pi)w)$$
$$x^- = \beta + \Phi^T(R - (\Phi - \gamma\Phi'^\pi)w) = 2\beta - x^+$$

Notice that both x^+ and x^- are in the range $[0, 2\beta]$. Furthermore, when an element $x_i^+ = 0$, it must be that $x_i^- = 2\beta$ (and vice-versa). With these definitions in hand, we can write the optimality conditions as:

$$\begin{aligned} 0 < x_i^+ < 2\beta \;\;\&\;\; 0 < x_i^- < 2\beta \;\;&\Longrightarrow\;\; w_i = 0 \\ x_i^+ = 0 \;\;\&\;\; x_i^- = 2\beta \;\;&\Longrightarrow\;\; w_i \geq 0 \\ x_i^- = 0 \;\;\&\;\; x_i^+ = 2\beta \;\;&\Longrightarrow\;\; w_i \leq 0 \end{aligned}$$

Now decompose the weight vector w into positive (w^+) and negative (w^-) portions such that:

$$\begin{aligned} w &= w^+ - w^- \\ w^+, w^- &\geq \mathbf{0} \\ \text{if } w_i \geq 0, \text{then } w_i^+ \geq 0 \;\;\&\;\; w_i^- &= 0 \\ \text{if } w_i \leq 0, \text{then } w_i^- \geq 0 \;\;\&\;\; w_i^+ &= 0 \end{aligned}$$

When we convert the optimality conditions to a linear complementarity problem (LCP), we will show that a solution to the LCP must satisfy these constraints placed on w^+ and w^-. The optimality conditions can now be written as:

$$\begin{aligned} 0 < x_i^+ < 2\beta \;\;\&\;\; 0 < x_i^- < 2\beta \;\;&\Longrightarrow\;\; w_i^+ = w_i^- = 0 \\ x_i^+ = 0 \;\;\&\;\; x_i^- = 2\beta \;\;&\Longrightarrow\;\; w_i^+ \geq 0 \;\;\&\;\; w_i^- = 0 \\ x_i^- = 0 \;\;\&\;\; x_i^+ = 2\beta \;\;&\Longrightarrow\;\; w_i^- \geq 0 \;\;\&\;\; w_i^+ = 0 \end{aligned}$$

Now it is easy to show these optimality conditions define a LCP. To help see this, note that (1) all the variables (x^+, x^-, w^+, w^-) must be nonnegative and (2) the only way a weight w_i^+ (or w_j^-) can be positive is if the corresponding variable x_i^+ (or x_j^-) equals zero. Using the notation $A = \Phi^T(\Phi - \gamma\Phi'^\pi)$ and $b = \Phi^T R$, the LCP is written:

$$\begin{aligned} \begin{bmatrix} x^+ \\ x^- \end{bmatrix} &= \left(\beta - \begin{bmatrix} b \\ -b \end{bmatrix}\right) + \begin{bmatrix} A & -A \\ -A & A \end{bmatrix} \begin{bmatrix} w^+ \\ w^- \end{bmatrix} \\ \begin{bmatrix} x^+ \\ x^- \end{bmatrix}, \begin{bmatrix} w^+ \\ w^- \end{bmatrix} &\geq \mathbf{0} \\ x_i^+ w_i^+ &= 0 \;\;\forall i \\ x_i^- w_i^- &= 0 \;\;\forall i \end{aligned}$$

Notice that a solution to the LCP must meet the constraints we placed on w^+ and w^-. To see this, assume $w_i^+ > 0$. If this is true, then x_i^+ must equal 0 (by complementarity) and $x_i^- = 2\beta$ (by definition) which in turn implies $w_i^- = 0$ (by complementarity). The same logic applies if $w_i^- > 0$.

Finally, note that the solution above is equivalent to performing the transformation in section A.1 to the BLCP formulation of the L_1 regularized fixed point.

A.3 The LARQ algorithm

The LARQ algorithm (Figure 2) modifies LARS-TD to identify policy change points in addition to feature change points. These policy change points, much like the feature addition/removal points of LARS-TD, can be found analytically. Policy changes can be made at a step size given by the equation

$$\alpha = \min_{a,i} -\frac{\Phi_i'^a \mathbf{w} - \Phi_i'^\pi \mathbf{w}}{\Phi_i'^a \Delta\mathbf{w} - \Phi_i'^\pi \Delta\mathbf{w}}, \tag{6}$$

where $\Delta\mathbf{w}$ is the vector along which the coefficients are currently constrained to move according to the LARS-TD invariants.

Algorithm LARQ ($\{s_i, a_i, r_i, s'_i\}, \gamma, \{\varphi_j\}, \beta$)

parameters:

$\{s_i, a_i, r_i, s'_i\}$ state, action, reward, and nextstate samples
γ : discount factor
$\{\varphi_j\}$: basis functions
β : regularization parameter

initialization:

$\mathbf{w} \leftarrow \mathbf{0}$
$\boldsymbol{\pi} : \pi(s'_i) \leftarrow 1$
$\boldsymbol{\Phi} : \Phi_{ij} \leftarrow \varphi_j(s_i, a_i)$
$\mathbf{c} \leftarrow \boldsymbol{\Phi}^T R$
$\{\bar{\beta}, i\} \leftarrow \max_j |c_j|$
$\mathcal{I} \leftarrow \{i\}$

while $\bar{\beta} > \beta$:

Get features for next states following policy $\boldsymbol{\pi}$:
$\boldsymbol{\Phi'}^{\pi} : \Phi'^{\pi}_{ij} \leftarrow \varphi_j(s_i, \pi(s_i))$

Find update direction:
$\Delta\mathbf{w}_{\mathcal{I}} \leftarrow (\boldsymbol{\Phi}_{\mathcal{I}}^T \boldsymbol{\Phi}_{\mathcal{I}} - \gamma \boldsymbol{\Phi}_{\mathcal{I}}^T \boldsymbol{\Phi}'^{\pi}_{\mathcal{I}})^{-1} \operatorname{sgn}(\mathbf{c}_{\mathcal{I}})$

Find step size for adding a feature to the active set:
$\mathbf{d} \leftarrow (\boldsymbol{\Phi}^T (\boldsymbol{\Phi}_{\mathcal{I}} - \gamma \boldsymbol{\Phi}'^{\pi}_{\mathcal{I}}) \Delta\mathbf{w}_{\mathcal{I}}$
$\{\alpha_1, i_1\} \leftarrow \min^{+}_{j \notin \mathcal{I}} \left\{ \frac{c_j - \bar{\beta}}{d_j - 1}, \frac{c_j + \bar{\beta}}{d_j + 1} \right\}$

Find step size for removing a feature from the active set:
$\{\alpha_2, i_2\} \leftarrow \min^{+,0}_{j \in \mathcal{I}} \left\{ -\frac{w_j}{\Delta w_j} \right\}$

Find step size to first of {add a feature, remove a feature, terminate at fixed point}:
$\alpha \leftarrow \min\{\alpha_1, \alpha_2, \bar{\beta} - \beta\}$

// This next section is omitted in LARS-TD
Find step size for greedy policy update
$Q'^{\pi} \leftarrow \boldsymbol{\Phi'}^{\pi}_{\mathcal{I}} \mathbf{w}_{\mathcal{I}}$
$\Delta Q'^{\pi} \leftarrow \boldsymbol{\Phi'}^{\pi}_{\mathcal{I}} \Delta\mathbf{w}_{\mathcal{I}}$
foreach action a
$Q'^{a} : Q'^{a}_i \leftarrow \Phi(s'_i, a)_{\mathcal{I}} \mathbf{w}_{\mathcal{I}}$
$\Delta Q'^{a} : \Delta Q'^{a}_i \leftarrow \Phi(s'_i, a)_{\mathcal{I}} \Delta\mathbf{w}_{\mathcal{I}}$
$\{\alpha^a_3, i^a_3\} \leftarrow \min_i \left\{ -\frac{Q'^{a}_i - Q'^{\pi}_i}{\Delta Q'^{a}_i - \Delta Q'^{\pi}_i} \text{ such that } \Delta Q'^{a}_i > \Delta Q'^{\pi}_i \right\}$
$\{\alpha_3, \pi'_3\} \leftarrow \min_a \{\alpha^a_3\}$
$i_3 \leftarrow i^{\pi'_3}_3$
$\alpha \leftarrow \min\{\alpha, \alpha_3\}$
// End of LARQ-only section

Update weights, $\bar{\beta}$, and correlation vector:
$\mathbf{w}_{\mathcal{I}} \leftarrow \mathbf{w}_{\mathcal{I}} + \alpha \Delta\mathbf{w}_{\mathcal{I}}$
$\bar{\beta} \leftarrow \bar{\beta} - \alpha$
$\mathbf{c} \leftarrow \mathbf{c} - \alpha\mathbf{d}$

Update active set or policy:
if $(\alpha = \alpha_1), \mathcal{I} \leftarrow \mathcal{I} \bigcup \{i_1\}$
else if $(\alpha = \alpha_2), \mathcal{I} \leftarrow \mathcal{I} - \{i_2\}$
else if $(\alpha = \alpha_3), \pi(i_3) \leftarrow \pi'_3$
end while
return $\mathbf{w}$.

Figure 2: The LARQ algorithm.

To satisfy the greedy invariant of LARQ, however, we must also ensure that the new policy we switch to will be better than the old policy as we move towards the new fixed point. In general, all that is necessary to satisfy this requirement is that the gradient of the new policy, $\Phi'^{\pi'}_i \Delta\mathbf{w}$ with respect to the direction of travel is greater than the gradient of the old policy.

A.4 LCP Solver

In our experiments, we used a modified complementary pivoting algorithm to solve for the L_1TD fixed point using the LCP formulation as shown at the end of appendix section A.1. We adapted a Matlab LCP solver due to P. Fackler and M. Miranda that allows for warm starts. The file `lemke.m` can be found in the CompEcon toolbox at `http://www4.ncsu.edu/~pfackler`. The code was adapted so that the full matrix $\Phi^T(\Phi - \gamma\Phi'^{\pi})$, which can be very large depending on the size of the feature set, was never explicitly formed. The full matrix is not needed to determine which element should be pivoted in and out of the active set at any iteration of the algorithm. We only formed the principal submatrix associated with the active elements. As a further optimization, rather than forming the principal submatrix itself, it is possible to update and downdate the *inverse* of this principal submatrix incrementally to solve the necessary linear system of equations.

In section 2.3, we stated that the LCP formulation is twice as large as the BLCP formulation. We used the LCP formulation here because the code was readily available and appeared robust. However, any suitable solver could be used instead.

A.5 Robustness of LARS-TD

When the A matrix is a P-matrix, LC-TD and LARS-TD are guaranteed to produce solutions that achieve the L_1TD optimality conditions. When the P-matrix property does not hold, both algorithms have the potential to violate the optimality conditions. In practice, we found LARS-TD to be much more susceptible to this problem. We discuss here a possible rationale explaining this behavior.

There are two ways that LARS-TD can violate the invariants: (1) the sign of the correlation may disagree with the sign of the weights for features in the active set, and (2) once a feature is removed from the active set, the absolute value of its correlation may in fact increase compared to the correlation of features in the active set. Both of these problems are of course detectable while running the algorithm. Instead of terminating prematurely when one of these two problems arises (in which case the algorithm would *not* return a set of coefficients for the desired value of the regularization parameter β), our LARS-TD implementation simply ignores the deviations and continues. This decision was made for practical reasons. We found it better for LARS-TD to return a set of coefficients at the requested value of β than to terminate early and return coefficients at a larger value of β. However, these LARS-TD violations did impact performance. For example, for one setting of β in our mountain car experiments, we found that the final policy learned using LC-TD reached the goal in 19 out of 20 trials, taking 136 ± 22 steps on average. The policies learned using LARS-TD reached the goal in 17 of 20 trials, taking 161 ± 41 steps on average.

The fact that LARS-TD produced solutions violating the L_1TD optimality conditions explains the empirical behavior shown in Figure 1. In those plots, notice that policy iteration converged after six rounds when using LC-TD whereas it did not converge at all when using LARS-TD. Recall that convergence is obtained when the coefficients w do not change significantly from one round to the next. Since LARS-TD was frequently violating the optimality conditions, it was not reaching an L_1TD fixed point whereas LC-TD was doing so. Thus, LC-TD was able to produce the same set of coefficients in two subsequent rounds of policy iteration. LARS-TD would violate the optimality conditions in different ways in subsequent rounds of policy iteration and therefore never stabilized on one set of coefficients.

We offer the following explanation as to why a homotopy method like LARS-TD may be more prone to violating the optimality conditions in comparison to an LCP solver. Consider the behavior of both LARS-TD and LC-TD for a fixed Φ, Φ'^{π}, R, and γ but a varying β. For a particular non-P-matrix $\Phi^T(\Phi - \gamma\Phi'^{\pi})$, there will exist at least one R such that there is not a unique solution achieving the L_1TD optimality conditions. As demonstrated by Kolter and Ng [1] (Figure 2b), changes in the number of solutions as β varies can correspond to discontinuities in the homotopy path. If LARS-TD encounters such a point, then it will fail to preserve the invariants when it crosses the discontinuity and it is unlikely to re-establish them. This is because the algorithm does not make the sort of discontinuous changes needed to jump onto a different homotopy path. In contrast, LC-TD directly solves the problem for a particular β (or perhaps a finite set of β values). Discontinuities in the homotopy path have no direct impact on the LCP solver's ability to find a solution. Moreover, if solutions are desired at a set of β values, a failure to find a solution for any particular value will not taint the solver's search for a valid solution at other values, even if using warm starts. In summary, a

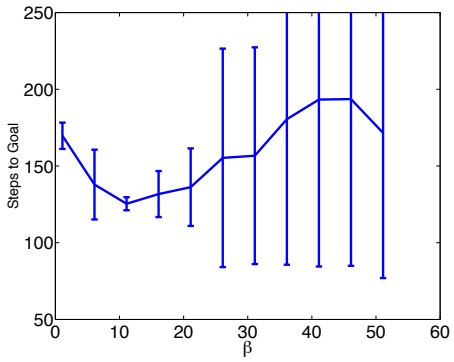

Figure 3: Average performance of final policy produced by LC-MPI for the mountain car MDP at different values of the regularization parameter β.

negative aspect of a homotopy method is that it can be derailed if a problem occurs at any point along the continuous homotopy path, while a direct method that searches for a solution at a particular value of the regularization parameter is influenced only by the properties of the solution at that particular value.

A.6 Use of Cross-Validation

Lastly, to show how one may use the results of LC-MPI, Figure 3 shows the average performance of the 11 greedy policies for mountain car. The graph shows the number of steps to reach the goal up to a maximum of 350 steps per trial. The large error bars for large values of β are due to poor policies resulting in tests that do not reach the goal within 350 steps. Notice the classic "bathtub" shape in which under- and over-regularized solutions perform poorly.