Theory and Simplex Implementation
Summer 2017: Progress Report for Week 8

Jonathan Robert Perlstein
jrperl{at}cs.stanford.edu

August 20, 2017

1 Simplex Implementation

Current Status My initial implementation of the simplex algorithm is complete. Please see the `simplex.pdf` document in the previous bundle for details concering this implementation. At this stage my program is able to solve linear programs of the form `max` $c^T x$ `st` $Ax \leq b$ via a generic simplex vertex-hopping methodology. My program also is able to find the optimal objective function value for linear programs of the form `min` $y^T b$ `st` $A^T y \geq c$. What I have implemented is plain simplex: I have not yet completed any generic simplex optimizations or any problem-specific optimizations.

Upcoming Work In order to optimize the program for this problem, I will implement the following upgrades.

- I am in the process of porting the tableau solving functionality to a fast compiled language. Given the memory constraints of our problem, this port should allow the $n = 4$ case to work very quickly. Once the $n = 4$ case is complete during the Fall quarter, I will evaluate whether porting at an earlier point in the simplex implementation [before the tableau] is needed for the $n = 5$ case.
- I am working with Tom to understand how to apply simplex directly to our problem. During this initial port, I will keep the same general output that the non-optimized version already has, which includes printing the tableau at each stage. As I gain greater understanding of the vertex-hopping aspect of our problem, I will refine the tableau output accordingly.

2 Theory: Simulation

2.1 Overview of Objective for Summer Quarter

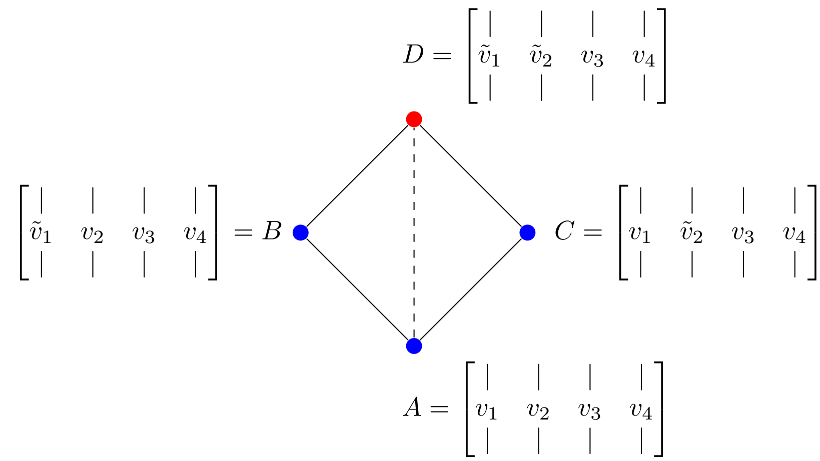
Prior to my work, previous work by the authors of *Blind Joint MIMO Channel Estimation and Decoding* [hereinafter: *Blind Decoding*] had provided mathematical justification for why `Algorithm 2` from the paper always converges when $n \in \{2, 3, 4\}$. Furthermore, the *Blind Decoding* authors had discovered a counterexample for the $n = 6$ case, leaving only the $n = 5$ case incomplete. Simulation performed by the *Blind Decoding* authors prior to my work had provided convincing empirical evidence that `Algorithm 2` indeed does always converge in the $n = 5$ case. My task was build on the tools that the authors used to in proving the $n = 4$ case and apply similar methodology to the $n = 5$ case. This section describes both the theoretical aspects of my work and provides a brief overview of the simulation code I developed to make the proofs possible.

The critical figure from *Blind Decoding* is reproduced here as Figure 1 [in *Blind Decoding* this is `Figure 3`]. Mathematical work done by the authors of the *Blind Decoding* demonstrates that, if $det(A) = det(B) = det(C) < det(D)$, and all matrices connected by solid lines are at Hamming distance one from each other, then the objective function increases monotonically along the line AD. Therefore, matrix A cannot be a local optimum.

The authors of *Blind Decoding* previously had demonstrated that, in the $n = 4$ case, for all matrices A such that $det(A) = \pm 8$, there is a matrix D such that $det(D) = \pm 16$ [where the sign of A and D are the same] and the Hamming distance between A and D is at most 2. Using this fact and Figure 1, the *Blind Decoding* authors showed that, in the $n = 4$ case, no matrix of determinant ± 8 can be a local optimum, and thererefore there are no local optima.

My primary task for the theory aspect of my Summer quarter work was to prove that Figure 1 holds in the $n = 5$ case as well.

Figure 1: Figure from *Blind Decoding*



2.2 Initial Objective

My first goal was to prove that, in the $n = 5$ case, each matrix A where $det(A) = 16$ has some matrix D such that $det(D) = 32$ and $hd(A, D) \leq 2$.[1] I concurrently attempted to prove that for each matrix A where $det(A) = 32$, there is some matrix D such that $det(D) = 48$ and $hd(A, D) \leq 2$. I solved this problem by writing an optimized de-facto graph traversal implementation, and I proved that both of the above conjectures do hold. My report of this is described in the `week1.pdf` document. The specific case of showing that each A such that $det(A) = 32$ is in fact immediately adjacent to some D where $det(D) = 48$ is provided in `week4.pdf`. The initial code is contained in `dist.c` [note the same algorithm is also implemented in `dist.rs`].

2.3 First Step: Show Figure 1 for $n = 4$ Case

To develop my simulation code on a manageable test case, I started by showing that Figure 1 holds in the $n = 4$ case. Specifically this involves matrices A where $det(A) = 8$ and closest matrix D such that $det(D) = 16$ is at Hamming distance 2 from A. Completing the demonstration requires showing that for each such A, there exist matrices B, C, D such that all of the following hold:

- $det(B) = 8$ and $det(C) = 8$
- $hd(A, B) = 1$ and $hd(A, C) = 1$
- Both of B, C are adjacent to the same matrix D [so $hd(B, D) = hd(C, D) = 1$] where $det(D) = 16$.

To demonstrate that Figure 1 holds in the the $n = 4$ case, I expanded my code to do the following:

- Let `highdet` be the determinant of the matrix D in Figure 1, and let `lowdet` be the determinant of the matrices A, B, C in Figure 1. Thus in the $n = 4$ case `highdet` $= 16$ and `lowdet` $= 8$.
- Create an empty vector for each node of determinant `lowdet`. Then collect all nodes of determinant `highdet`. For each matrix of determinant `highdet`, enumerate every neighbor [node at Hamming distance 1] of determinant `lowdet`. For each such neighbor, add the matrix of determinant `highdet` to the vector corresponding to the matrix of determinant `lowdet`. Thus, at the end of this process, for every matrix of determinant `lowdet`, we have a list of all matrices of determinant `highdet` that are adjacent to it.
- For each node of determinant `lowdet` that is not adjacent to *any* matrix of determinant `highdet` [these are the matrices that are 2 hops away from the closest matrix of determinant `highdet`], verify that the graph shown in Figure 1 holds. Specifically, if we let A be the matrix of determinant `lowdet` that is not adjacent to any matrices of determinant `highdet`, then there must be *some* pair of matrices B, C that are adjacent to A, also have determinant `lowdet`, and are themselves adjacent to the *same* matrix D of determinant `highdet`. In terms of implementation, at this point in the program every matrix of determinant `lowdet` has a list of matrices of determinant `highdet` to which it is immediately adjacent. For every matrix of determinant `lowdet` for which this list is empty [so the closest matrix of determinant `highdet` is two hops away, and this matrix plays the role of A in Figure 3], I examine all of matrix A's neighbors to verify that there is *some* combination of two neighbors [these play the role of B and C

[1] I use the abbreviation *hd* to stand for Hamming distance.

in Figure 3] that have at least one matrix of determinant `highdet` to which *both* B and C are adjacent [this is the matrix D in Figure 1].

Through this methodology I demonstrated that Figure 1 holds for all matrices A where $det(A) = 8$ in the $n = 4$ case.

2.4 Expansion to $n = 5$ Case

2.4.1 Initial Counterexample

I initially attempted to port this idea directly to the $n = 5$ case, showing that Figure 1 holds for all matrices A where $det(A) = 16$. In my first attempt, I allowed $det(D) = 32$ since this seemed like the clearest analogue to what I had shown in the $n = 4$ case. I quickly found a counterexample. Specifically, I found a matrix A such that all of the following hold:

- $det(A) = 16$.
- The closest matrix D such that $det(D) = 32$ is at Hamming distance 2 from A.
- Among the set of all neighbors of A that have determinant 16, thus all the matrices that could play the role of B and C, there is no pair of matrices B, C such that both B and C are adjacent to the *same* matrix D where $det(D) = 32$.

The counterexample is detailed below:

- Matrix: 00008890
- Neighbors B such that $det(B) = 16$ and B is one hop from at least one matrix D such that $det(D) = 32$. All such candidate matrices D are listed as well.
  - Neighbor: `00008891` candiate matrices D = [`00808891`].
  - Neighbor: `00008892` candiate matrices D = [`00808892`].
  - Neighbor: `00008894` candiate matrices D = [`00808894`].
  - Neighbor: `000088B0` candiate matrices D = [`008088B0`].
  - Neighbor: `000088D0` candiate matrices D = [`008088D0`].
  - Neighbor: `00008A90` candiate matrices D = [`00808A90`].
  - Neighbor: `00008C90` candiate matrices D = [`00808C90`].
  - Neighbor: `00009890` candiate matrices D = [`00809890`].
  - Neighbor: `0000C890` candiate matrices D = [`0080C890`].
  - Neighbor: `00018890` candiate matrices D = [`00818890`].
  - Neighbor: `00028890` candiate matrices D = [`00828890`].
  - Neighbor: `00088890` candiate matrices D = [`00888890`].

2.4.2 Revised Methodology

I then expanded the methodology for the $n = 5$ case. Instead of allowing matrices playing the role of D in Figure 3 to be solely matrices of determinant 32, I expanded the whole idea of a `highdet` matrix to be any matrix of determinant in $\{32, 48\}$. The implementation of the change is as follows.

- Any matrix A where $det(A) = 16$ that is immediately adjacent to a matrix of determinant 32 *or* 48 is automatically not a local optimum.
- The matrix that plays the role of D can have determinant *either* 32 or 48, and the same demonstration from Figure 3 of the paper will hold. To test whether this expansion will allow the $n = 5$ case to work completely, I modified the program as follows:
  - I essentially view `highdet` to be the set $\{32, 48\}$. Thus, when I take each matrix of determinant `highdet` and enumerate all immediate neighbors of determinant `lowdet`, I am taking each matrix of determinant in the set $\{32, 48\}$ and adding that matrix to the vector corresponding to every neighboring matrix of determinant 16. This idea of `highdet` being a set rather than a single value holds throughout the analysis.
  - In effect, this simply means that the matrix playing the role of D in Figure 1 can have determinant either 32 or 48. The same logic used in the proof holds regardless of whether $det(D) = 32$ or $det(D) = 48$.

2.4.3 Result of Revised Methodology

Expanding the concept of `highdet` to be the set $\{32, 48\}$ in the $n = 5$ case allows the simulation to go through successfully. The code for this revised methodology is present in `dist.rs`.

3 Theory: Lemma for $n = 5$ Case

I have the following preliminary draft language pertaining to the $n = 5$ case.

Lemma: *For $n = 5$ all optima are global.*

We performed a computer simulation analogous to the $n = 4$ case, creating a graph with a node for each matrix A in $\{-1, +1\}^{5\times5}$ where $det(A) \neq 0$, with an edge between each pair of nodes with Hamming distance one. The possible nonzero determinants in the $n = 5$ case are $\{\pm16, \pm32, \pm48\}$. We show here that no matrices with determinant ±16 or ±32 are local optima. A basic graph traversal algorithm such as depth-first-search reveals that, in the $n = 5$ case, the graph consists of a single connected component. If all matrices of determinant ±16 are removed from the graph, then the remaining graph separates into two connected components similar to the $n = 4$ case: one component contains all matrices with positive determinant $\{+32, +48\}$, and the other component contains all matrices with negative determinant $\{-32, -48\}$.

We show that no matrices with determinant $\{\pm16, \pm32\}$ are local optima. We specifically demonstrate that no matrices with determinant $\{+16, +32\}$ are local optima, which by symmetry implies that the same holds for matrices with determinant $\{-16, -32\}$. We have the following two observations, both of which were verified via computer simulation. Let $S = \{-1, +1\}^{5\times5}$, and let $hd(A, B)$ be the Hamming distance between matrix A and matrix B.

- Observation 1. $\forall\{A \in S \mid det(A) = 32\} : \exists D \in S :$

$$(det(D) = 48) \wedge (hd(A, D) = 1)$$

- Observation 2. $\forall\{A \in S \mid det(A) = 16\}$, one of the following two must hold:
  - First Possibility.
  $$\exists D \in S : det(D) \in \{32, 48\} \wedge hd(A, D) = 1$$
  - Second Possibility.
  $$\begin{aligned}&\exists B, C, D \in S :\\ &(det(B) = det(C) = 16) \wedge\\ &(det(D) \in \{32, 48\}) \wedge\\ &(hd(A, B) = hd(A, C) = hd(B, D) = hd(C, D) = 1)\end{aligned}$$

It follows immediately from Observation 1 that no matrix with determinant $+32$ can be a local optimum. From Observation 2, if the first possibility holds, then the given matrix A cannot be a local optimum. If the second possibility holds, then the same argument as Section V, Subsection D applies. Each such matrix A where $det(A) = 16$ is adjacent to two matrices B, C where $det(B) = 16, det(C) = 16$ which themselves are both adjacent to a matrix D such that $det(D) \in \{32, 48\}$. The same argument used in the previous subsection thus shows that the objective function is monotonically increasing along the line AD; therefore, A is not a local optimum.

4 Planned Further Work

For the intercessional period before the Fall quarter starts, I plan to do the following:

- Optimize the simplex implementation as described in Section 1.
- Gain a greater understanding of when optimal solutions to the linear program do and do not correspond to actual solutions of our problem.

I will provide updates concerning this work as I progress.