Author Response

We thank the reviewers for their valuable feedback.

Experiment comparing CD and CG *R2*: You make a great point that each iteration of CD uses two line searches whereas CG uses only one. For problems in which the cost of a line search is comparable to that of solving the subproblem, the iteration count is probably not the most fair metric with which to compare CD and CG. Because of this, in Figure 1, we compare the number of matrix multiplications used by CD and CG instead. At each iteration of CD and CG, the computational cost of line searches is dwarfed by that of the repeated matrix multiplications with $\mathcal{G}^*(\nabla f(\eta_k y_k + (\eta_k - 1)g))$ necessary to compute the minimum eigenvalue via the Lanczos algorithm. If it were the case that the additional line search at each iteration of CD were only saving one minimum eigenvalue computation, then we would expect that CD would use about half as many matrix multiplications as CG. However, the histogram indicates that CD empirically does better. We cannot prove why this occurs, but we speculate on line 230.

Experiment comparing CD and BM *R3*: The large dots on the left in Figure 2 show the performance of a traditional BM implementation at various ranks. We will annotate this directly on the plot to make it more clear. We see that pure BM, *i.e.*, without the safeguard, does not converge to the global optimum for $r = 2$, 3, 4 or 5. Even without the greedy heuristic, CD outperforms pure BM for $r = 2, 3, 4$, albeit after more matrix multiplications. At a high enough rank, pure BM will probably converge to the optimal value, but we are primarily focused on problems for which memory is the limiting factor.

Assumption of no nonzero direction of recession *R3*: We agree that the assumption that f have no nonzero direction of recession in K is cludgy. To guarantee this is satisfied in problem (8), we could add an extremely small trace penalty. This probably would not affect our numerical results in any way. As an alternative assumption, we could stipulate that the set of optimal points is bounded and nonempty. This would imply that there is no nonzero direction of recession of f in K.

Matrix sketching and memory efficiency *R1*: We feel that the theory and practical utility of randomized matrix sketches in an optimization algorithm have been well established already in the literature, *e.g.*, [10, 20, 22, 23], and due to the space limitations, we treat the sketches essentially as tools. We will improve the citations to the relevant literature in Section 4. In Appendix F, we will move the citations on line 453 directly to line 452. *R2*: The use of the sketch means that we do not retain the q_k at each iteration. The parameter in the sketch determines how much memory we use (only $3n$ numbers in the experiment in Figure 1). *R4*: In line 10 of Algorithm 2, we do not need to form $q_k q_k^T$. The operator $\mathcal{G}$ can be evaluated efficiently on rank-1 matrices because $\mathcal{G}(q_k q_k^T) = (q_k^T G_1 q_k, \ldots, q_k^T G_m q_k)$ and matrix vector products are efficiently computable for the G_i (by assumption). The same idea holds for line 7 of Algorithm 5. In our implementation, we make sure to exploit this.

Burer-Monteiro *R1*: It is fair to point out that CD is a heuristic. We should be clearer about that. The assumptions we mention in line 44 that are required in order to guarantee the global convergence of BM are very difficult to establish in practice. As far as we are aware, they are essentially the matrix version of the restricted isometry property (restricted strong convexity and restricted smoothness), and the rank necessary for the guarantees is frequently higher than that needed to specify the solution. Verifying the assumptions we make regarding problem (5) is relatively easy and can frequently be carried out by inspection. For memory-efficient CD, we really only need to check whether G_i have efficient matrix multiplication routines and how large m is relative to n^2.

Related works *R1*: We do not address CGAL in this paper because we only consider the cone constraint. However, we think that directly handling equality constraints is the next natural step for future work. The connection between OMP and greedy conic optimization has been examined in detail in [16], and unfortunately, we do not think that we have the space to cover it here adequately. *R3*: Thank you for bringing the paper by Journee et al. to our attention. We will include it in our introduction. *R4*: Regarding the Ochs et al. paper, we do not see how to make meaningful comparisons for large scale optimization over the PSD cone using any model function other than the one which reduces to CG, *i.e.*, a simple minimum eigenvalue computation at each iteration. Regarding the Rao et al. paper, in order to maintain memory-efficiency for very large n, we cannot retain the prior iterates necessary to run CoGEnt. Regarding the Braun et al. paper, caching the prior iterates as part of the weak separation oracle would also significantly increase the potential memory requirements.

Other *R1*: We will adjust our citations to reference published versions instead of ArXiv pre-prints whenever possible. Also, we will clean up the sentence on line 279. *R3*: We will add the citation to [22] in line 172. *R4*: The notation $\mathbf{dist}^*$ is defined on lines 83-84.