On the solution of large-scale SDP problems by the modified barrier method using iterative solvers

by Kocvara and Stingl

This paper is on the numerical implementation of the modified barrier method using iterative solvers. The main issues are on the evaluation of Hessian-vector products; the appropriate stopping criterion and preconditioning of the iterative method used to solve the Newton equations arising from the successive minimization of the augmented Lagrangian.

On the whole, I like this paper and feel that the paper contains materials that are of current interest and worth publishing. But I would like to see an improved version that addresses the points raised below.

Let me discuss the positive and negative points of the main contributions of the paper.

1. [The modification of the authors' code (PENNON) to accommodate the use of approximate Newton direction when the Newton equation is solved by iterative methods.]

   - I think the authors must have spent quite a lot of effort in tuning the various parameters in PEN-PCG to make the algorithm works. I do not expect theoretical justification for the heuristics used in algorithm. But it would be good if the authors can motivate the rationale behind the proposed heuristics. For example, in p.5 on the strategy for updating the penalty parameter, there is no motivation on the choice made in Steps 3,4, and 5.
   - The paper emphasized that the PCG takes very few iterations to solve the Newton equation at each Newton step, even when close to optimality. But there is no evidence to suggest that the preconditioners greatly reduce the conditioner number of the Hessian. If the preconditioned system is still ill-conditioned, then the low number of PCG iterations required must be the result of extremely loose stopping tolerance set for the solver. It is very surprising that such a loose stopping tolerance for computing the inexact Newton direction does not affect the overall convergence behavior of the modified barrier method. *It would be good if the authors can look into this issue more closely and suggest possible reasons for such a surprising phenomenon. I think the algorithm must be giving up something somewhere to allow it to make progress despite the use of loose inexact Newton direction. One possible question is that: if the semidefinite constraint,* $C(x) \preceq C_0$, *in (13) is strictly enforced, would the convergence behavior of the modified barrier method still remain unchanged?*

2. [The use of finite difference to approximate a Hessian-vector multiplication.]

- The authors used $\nabla^2 F(x_k)v \approx (\nabla F(x_k + hv) - \nabla F(x_k))/h$ with $h = (1 + \|x_k\|10^{-3})$. This seems to be a purely empirical choice. It does not relate to the existing literature (see for example, V. Simoncini and D.B. Szyld, SIAM J. Scientific Computing, 25 (2003), pp. 454-477; and J. Eshof and G. Sleijpen, SIAM J. Matrix Analysis and Applications, 26 (2004), pp. 125-153) on inexact Krylov methods where there are criterion (or guideline) given on the accuracy require in the approximate Hessian-vector product. *There should a discussion on the accuracy in (12) in relation to those suggested in the literature.*
- In evaluating $\nabla F(x_k + hv)$ for each new v, is the matrix $\mathcal{Z}(x_k + hv)$ recomputed? Or is it approximated by $h\mathcal{Z}(x_k)\mathcal{A}(v)\mathcal{Z}(x_k)$? Please mention this clearly in the paper.

3. [The evaluation of the performance of several preconditioners (diagonal, symmetric Gauss-Seidel, L-BFGS, and AINV) on the ill-conditioned Newton equations.]

   - Except for L-BFGS, this is a straightforward application of standard preconditioners to the Newton equations. Of the four, the symmetric Gauss-Seidel preconditioner should not be looked upon as a viable choice for large SDP because it required the dense Hessian to be computed explicitly, which is usually too expensive in terms of CPU time and memory. The AINV preconditioner is also not a viable choice because the inverse of the dense Hessian typically has elements that are of the same order of magnitude, and as a result, a loose drop-tolerance will result in a useless preconditioner, but the opposite will result in a dense matrix. The poor performance of AINV preconditioner is not too surprising when the drop-tolerance is chosen loosely to produce a sparse matrix.

     It is mentioned in 4.3 that the construction of the diagonal preconditioner is too costly because it requires n gradient evaluations. If the numerical experiments in Section 5 confirm this observation, then it is also not a viable choice, especially also because it is not a powerful preconditioner.

     This leaves only the L-BFGS preconditioner as the only promising choice in general. It is mentioned in Section 7 that the recent preconditioner proposed by Monteiro, O'Neal and Nemirovski (called MON preconditioner) performed very well in preliminary testing, *I would suggest including the MON preconditioner in the present paper. It will add more weight to the paper to include preconditioners that are viable rather than those that are quite obviously not suitable for large scale SDPs.*
   - The use of L-BFGS preconditioner for large scale SDP first appeared in [1]. The authors should reference and describe what has been done in that paper.

- In the numerical results in Section 5, it is quite clear that the CG method using preconditioners that required H explicitly is not competitive compared to the direct method in most cases. No doubt there are a few examples where explicitly computing H and using it within PCG is faster than the direct method, we know that these are problems where factorizing H is much more costly than computing H. Even then, when the size of H is large, such an approach would no longer be possible.

  Thus I would suggest excluding PCG(SGS) and PCG(AINV) altogether in the paper, but the authors can briefly mentioned that these preconditioners have been tested but are not competitive.

4. [Extensive numerical evaluation of the proposed iterative approach on 4 collections of SDPs.]
   The authors have done an impressive job in evaluating the performance of the proposed iterative approach.

   - However, the tests should concentrate on SDPs where there is not enough memory to store H. Those are the problems that can truely evaluate the performance of the proposed iterative approach. *That is, in the implementation, do not compute H explicitly but only accessed through its action on vectors. This will exclude preconditioners that require H explicitly.*
   - I presume that for PEN-PCG(diag), PEN-PCG(SGS), PEN-PCG(AINV), H first computed explicitly and then use within the CG iteration. Is the same done for PEN-PCG(BFGS)? This crucial information needs to be mentioned clearly in the paper.

5. The writing the paper is not very polished. There are quite a number of grammatical errors, and improper use of prepositions and incomplete sentences. I would not correct them but leave them to the editorial staff.

Below are some minor comments and points that need to be clarified.

[general] "criterium" should be "criterion".

[general] "per one Newton step" should be "per Newton step".

[general] "residuum" should be "residual".

[Section 5] Many of the table numbers referenced in the text are mixed up.

[p.2, bottom] "Further, the condition number of the Hessian itself is bounded close to the optimal point, provided..."

Please provide a reference where this statement is proved.

[p.3, ln 3] "In Section 4, we present examples **of**"

[p.3, notation] It seems that the subscript in $\langle A, B\rangle_{S_m}$ is not necessary.

[p.4, ln 4] The notation $D_{\mathcal{A}}\Phi_p(\mathcal{A}(x); U^k)$ is not clear. Is it the Frechet derivative? Should $\mathcal{A}(x)$ be $\mathcal{A}(x^{k+1})$?

[p.3, middle] "there exist $\bar{p}$ such that the minimum eigenvalue of the Hessian of the Lagrangian (3) is bounded away from zero ..."

Please provide a reference where this statement is proved.

[p.5, (8)] Should $\mathcal{Z}(x)$ be $\mathcal{Z}(x^{k+1})$?

[p.6, (10)] Should $f(x^k)$ etc be $f^T x^k$?

[p.6, middle] "$O(m^3n + m^2n^2)$ for dense matrices and $O(m^2n + K^2n^2)$ for sparse data matrices"

Why is m^3n changed to m^2n? What is the assumption made on the sparsity of the data matrices?

[p.7, middle] "it is well known that in finite arithmetics the actual number of CG iterations ... depends solely on the spectrum of the matrix H"

There are 2 points that is not entirely correct. First, the convergence of the CG method typically depends primarily on the spectrum of H, regardless of whether exact arithmetic or finite precision arithmetic. Second, its convergence can also depends on the right-hand side vector if the vector is special. Thus it is not correct to use the word "solely".

[p.7, middle] "grouping of eigenvalues"

It is more common to use "clustering of eigenvalues".

[p.10, ln -7] "we are close to machine precision and can hardly expect convergence"

This sentence is not clear. Please rewrite.

[p.12, 4.4] Please mention that the matrix H needs to be computed and stored explicitly to construct the symmetric Gauss-Seidel preconditioner.

[p.15, ln -8] "This ill-conditioning may be so severe that it does not allow ... within reasonable accuracy. Fortunately, this was not observed ..."

Please give some possible reasons on why the ill-conditioning not an issue in the present approach?

[p.16, middle] "Table 5.1"

Should it be Table 1?

[p.16, middle] "in few cases it is significantly faster. These examples (`theta*, and qap*`)"

This observation does not hold true for `qap`$*$.

[p.17, ln 3] "For the three codes, PENSDP, PEN-PCG(diag)"
This is inconsistent with the table. Should it be PEN-PCG(BFGS)?

[p.17, ln -7] "are typical by high ill-conditioning of the Hessian"
This sentence is not clear. Please rewrite.

[p.19, ln 5] "These problems are ill-conditioned; as a result"
Ill-conditioned in what sense? Numerical linear algebra sense or Renegar sense?

[p.21, 2nd paragraph] "Table 5.4" should be "Table 7"?

[p.21, 2nd paragraph] "From PENSDP to PEN-PCG(SGS) and further to"
This is inconsistent with the table. Should it be PEN-PCG(diag)?

[p.25, middle] "already the first iterations of the algorithm run with $\delta_{\mathrm{DIMACS}} = 10^{-1}$ differ from the run with $\delta_{\mathrm{DIMACS}} = 10^{-2}$, due to ..."
This sentence is not clear. Please rewrite.

[p.27, ln 6] "effect" should be "affect"?

[p.28, ln -4] "energetic norm"
What is this norm?

[p.30] Some references have years given, but some do not.

References

[1] M. Fukuda, M. Kojima, and M. Shida, Lagrangian dual interior-point methods for semidefinite programs, SIAM Journal on Optimization 12 (2002) 1007-1031.