

progress on MG/OPT

Zichao Di

May 2, 2011

1 Pieces of Algorithm:

- *Tried Initial estimate:*
 - random numbers
 - close-solution variables
- *Pre-smoothing:* Use fmincon to compute $(\bar{y}_h, \bar{\lambda}_h)$.
- *Downdate:* $\bar{y}_H = I_h^H \bar{y}_h$ where I_h^H is the full weighting matrix starting from the second variable
 $\bar{\lambda}_H = I_h^H \bar{\lambda}_h$
 $\hat{V}_H = \nabla_y \hat{L}_H(\bar{y}_H, \bar{\lambda}_H) - I_h^H \nabla_y \hat{L}_h(\bar{y}_h, \bar{\lambda}_h)$
 $\hat{S} = C_H(\bar{y}_H) - I_h^H C_h(\bar{y}_h)$

Remark: Choosing such downdate operator is reasonable since it satisfies the variational property, and also, the surrogate constraint is consistent with the analytic conclusion that fine and coarse level share same bound constraints under full weighting.
- *Shifted model:* $g_s(y_H) = g_H(y_H) - \hat{V}_H^T y_H$ s.t. $C_s(y_H) \equiv C_H(y_H) - \hat{S} \leq 0$ which in our case is $x \geq -0.01$.
- *OPT* $g_s(y_H)$ by fmincon get y_H^+
- $e_H = y_H^+ - y_H^-$, $e_h = I_h^h e_H$ where $I_h^h = 2[I_h^H]^T$, I believe I_h^h is chosen right too since except the same reason for I_h^H . Also, with the above update and downdate operator, MG/OPT works well for unconstrained problem

- *Line search:* Use Armijo based on Merit function to find the step length.

Issues here:

- *I can't apply merit function to MG-TN since I can't compute the corresponding λ*
- *be able to apply merit function to MG-Fmincon, but having trouble to compute the gradient GM of merit function.*¹
- *post-smoothing:* $y_h^+ = y_h^- + \alpha e_h$, apply fmincon.

2 All pieces of fmincon:

$$\min f(x) \text{ s.t. } \begin{cases} (x) \leq 0 \\ \text{ceq}(x) = 0 \\ Ax \leq b \\ Aeqx = beq \\ lb \leq x \leq ub \end{cases}$$

The way to apply fmincon: $x = \text{fmincon}(\text{fun}, x_0, A, b, Aeq, beq, lb, ub, \text{nonlcon}, \text{option})$.

In our case, given $Aeq, beq, lb, ub = [\]$, $A = -I_{n \times n}$, $b = 0.01e_n$ where e_n is the vector every component = 1.

Algorithm: Interior-point method.²

Issues:

- *Merit function used in fmincon is different from our fmincon, does it matter?*
- *The first-order optimality condition used in fmincon $\nabla F = \|v * g\|_\infty$, where v is defined in box-constraint, I tried apply exact solution as initial estimate into fmincon with one-bound constraint, it doesn't improve convergence. Conversely at the first iteration, fmincon push away the solution to a wrong direction. i.e. the first output ∇F is different away from the corresponding optimal ∇F^* . Theoretically, $\nabla F^* \approx \nabla F$, where in our case, ∇F has magnitude 10^2 .*³
- λ^k get from opt-con doesn't satisfy complementary condition, since in the unbinding component of λ^k , it should equal to 0 but in fact, it has magnitude 10^{-3} . According to section 15.3 [Giva], it makes sense since $\|x^k - x^*\| = O(\epsilon) = \|\lambda^k - \lambda^*\|$.

¹Note: We choose merit function as $M(x) = f(x) + \text{violation}^T \lambda + \frac{\rho}{2} \text{violation}^T \text{violation}$, where $\text{violation} = \max(Ax - b, 0)$, ρ , penalty parameter, tried $[1, 10, 100]$ doesn't change result very differently, GM is used to test if the direction e_h is a descent direction by $e_h^T GM < 0$

²Note: In fmincon, we don't really need to provide λ since it will compute automatically.

³Note: This is the most important issue I have to explore

3 Reasons MG-fmincon should work:

- *It works well for unconstrained case*
- *The coarse level solution is a good approximation to the fine level solution.*⁴
- $I_h^H = C[I_H^h]^T$ and $I_h^H A_h I_H^h = A_H$.

4 unsure elements

- *Is that enough to set the coarse constraint same to fine level? No, since it has been proved using current downdate will distort direction from coarse level*
- *Coarse level correction doesn't give a sufficient correction to fine level*⁵

5 Some extra experiments I have tried

- Pull out the coarse shifted problem, treat as a single optimization problem byt OPT_TN and test if it converges well.
 - Conclusion: it is converging well, so form this aspect, our coarse model should be built correctly.
 - Hint: the problem should be most possibly at transfer operator or the choice of step length.
- Treat the constraint as bound constraint in 'fmincon'. So there is no linear and nonlinear constraints any more. Optimize on it on single level, it did converge to the correct solution. But apply with MG ,the convergence failed for the same output that at some point, the variables get stuck and don't move any more.
- output the variable after everytime 'fmincon' called 'sfun' in order to know how does 'fmincon' treat infeasible points

6 Progress on Apr.1

- realize some weird behavior of fmincon doing on the infeasible variables, it may have a specific way to direct the infeasible point to feasible ⁶

⁴Note: 'refine' works well

⁵Note: step solution is inconsistent with search direction

⁶Guess: interior-point method in fmincon may have a general choose of step length to push infeasible point to feasible region which result in too far from exact solution and cost more to get back to after pre-smoothing .

- Try to manually fix the infeasible variable to strictly feasible in line search of MG/OPT so that when call fmincon to post-smoothing, fmincon won't waste computation to pull back infeasible to feasible. ⁷

7 Progress on Apr. 13

- For constrained case with fmincon, fix the mistake when computing the gradient of merit function at line search, and the performance is improved which at the first several iterations, MG/OPT showed much benefit rather than OPT by fine level evaluation gradient, although after that, MG/OPT-fmincon installed at some point because the fmincon set up will still pull the infeasible point too far back to strictly feasible region. But we agreed that this performance is enough to show the improvement of MG/OPT.
- For bound constraint with TN, I applied merit function to line search and also let Lagrangian function come into play at the step to construct surrogate model by using similar least square lagrangian multiplier estimate to estimate the multipliers at the current point. Similar consequence to fmincon, MG/OPT-TN also shows benefit at the first few iterations and then installed, the reason is the downdate operator pulls the coarse binding solution to unbinding and give a negative decent direction to the already binding fine solution.

8 Next steps:

- *Apply the coarse constraints setting up by Ph.L.Toint to our algorithm.*
- Once done the one side bound constraint, try two sides bound constraint and do the same test.
- Figure out how fmincon estimate lagrangian multiplier at the current solution.
- *Explore more on fmincon. eg. how it computes the 1st-order optimality, how it restrict variables in feasible region?*
- *Simplify my MG-fmincon, double-check on every little piece to make sure no mistake.*⁸

⁷Improvement: In this not general fixing way, $MG/OPT_{fmincon}$ does take advantage from coarse optimization

⁸Note: all pieces including the unrelavant part with current case

9 Goals:

- First critical thing is to figure out why two variables with tiny difference⁹ will show dramatic difference¹⁰ in the first iteration of 'fmincon'¹¹
- *For unconstraint case, improve fmincon to make it have the same performance as TN*
- *For constraint case, improve MG-Fmincon to make it converge. Once this is done, may have some hint how to fix MG-TN*
- *Once it converges, prepare the performance with OPT*

10 Tasks for the coming weekend

- 2D-CVT:
 - Run MG/OPT on 2D-CVT triangle domain
 - double check how does the code cut on the out of bound generators
- CVT paper:
 - Replot pictures on cycles, convergence factor for more generators with same low perturbation.
 - Plot the same element comparison for Lloyd 2D-CVT.
 - Test more random initial approximation on 1D-CVT
- Finish homework on Measure Theory and take a look of Thursday's note on Evolution Equation

11 Plan for next week

- Constrained MG/OPT:
 - Start to learn more on Interior -point method and measure function.
 - Try MG/OPT more cycles with modified line search

⁹Note: this difference is in terms of infinity norm of their subtraction

¹⁰Note: This difference is in terms of the output function value

¹¹Note: This observation is found at the beginning of post-smoothing which take the value from recursion as initial guess with 2 levels