NONLINEAR PROGRAMMING ALGORITHMS FOR LARGE NONLINEAR GASOLINE BLENDING PROBLEMS

Maame B. Poku, Lorenz T. Biegler
Carnegie Mellon University
Pittsburgh, PA 15123

Jeffery D. Kelly
Honeywell Hi-Spec Solutions
300 Yorkland Blvd.Toronto, Ontario, M2J 1S1Canada

Ronald Coxhead
Honeywell Hi-Spec Solutions
Bahnhofstrasse 44/46,D-65185 Wiesbaden, Germany

Vipin Gopal
Honeywell Laboratories
3660 Technology Drive, Minneapolis, MN 55418

Abstract

Gasoline production often yields 60-70% of a typical refinery's total revenue. A tight control of blending operations, a key step in gasoline production, therefore provides a crucial edge to the profitability of a refinery. A generalization also known as the pooling problem is used to model many systems with intermediate mixing (or pooling) tanks in the blending process (Audet et al., 2000). The classical blending arises in refinery processes where feeds with different quality attributes (sulfur composition, density or octane number, or boiling point temperatures) and flow rates are mixed to obtain products of desired qualities. Currently many blending applications are treated as extensions of linear blending models. Successive linear programming (SLP) strategies are applied to handle the nonlinear elements, but have shortcomings in terms of robustness and time of convergence. In this study we will compare and analyze numerical results of several large-scale gasoline blending models using current Nonlinear Programming (NLP) solvers LANCELOT, MINOS, SNOPT, KNITRO, LOQO and IPOPT. Although qualitative arguments will be made, a numerical comparison to a SLP code will not be presented.

Keywords

Bilinear programming, Gasoline blending, Pooling problem.

Introduction

Gasoline blending problems represent large-scale multiperiod nonlinear programs with mass balance constraints, nonlinear blending properties, large-scale structure (particularly across multiperiods) and combinatorial aspects dealing with switching strategies etc. In addition, blending systems are often encountered in other process industries e.g., chemical, pharmaceutical, cosmetics and food.

Gasoline blending models often include nonconvex nonlinearities, which lead to the existence of several locally optimal solutions (Adhya and Sahinidis, 1999). Given the high volumes of sales of petroleum products, the global optimization of the pooling and blending process could lead to substantial savings in cost, resulting in higher profit margins. Local solutions can also lead to significant improvements and they can be generated much faster, even for blend planning and scheduling applications. Moreover efficient local solvers are often necessary components of a global optimization algorithm.

Currently many blending applications are treated as extensions of linear blending problems. Successive linear programming (SLP) strategies are applied to handle the nonlinear elements, but have shortcomings in terms of robustness and convergence. In particular, global convergence for SLP can be achieved with trust region strategies. However, quadratic convergence for SLP is only possible with vertex optima. Otherwise, the convergence rate is at best linear and is dictated entirely by adjusting the trust region, which must shrink to zero at the solution. On the other hand, conventional large-scale nonlinear programming (NLP) strategies (like SNOPT and MINOS) may not be well suited for these problems for the following reasons. First, they are geared to optimization problems with few degrees of freedom, also known as superbasic variables (Gill et al., 1981); blending problems may have many superbasic variables. Secondly, since they approximate second order information using quasi-Newton updates, the number of iterations for the NLP solver frequently grows polynomially with problem size. Lastly, these NLP approaches do not provide a straightforward extension to handle discrete combinatorial elements in blending. To overcome these limitations, we consider a novel full space barrier (or interior point) approach for this nonlinear problem also known as NLP solver IPOPT. In addition, a novel filter line search strategy ensures convergence of the barrier problem. In this study we also compare and analyze numerical results for several large-scale gasoline blending models using the NLP solvers LANCELOT (Conn et al., 1992), MINOS (Murtagh and Saunders, 1993), SNOPT (Gill et al., 1997), KNITRO (Nocedal et al., 2000), LOQO (Vanderbei and Shannon, 1999), and IPOPT (Waechter, 2002).

AMPL a mathematical programming language, which provides automatic generation of first and second order derivatives of the Lagrangian function for the nonlinear problems, will be the interactive environment for solving these mathematical programming problems (Fourer et al., 1993).

Information on the NLP solvers used

Five of the various NLP solvers currently available (KNITRO, LANCELOT, LOQO, MINOS, and SNOPT) were chosen in order to compare the relative efficiency of IPOPT to the other solvers. A brief summary of each of the NLP solvers will be given.

In order to simplify the presentation of the IPOPT algorithm we assume that all variables have lower bounds of zero. An assumption is made that the optimization problem (NLP) can be stated as:

$$\begin{aligned} &\min f(x) \\ &s.t.\ c(x) = 0 \\ &\quad x \geq 0 \end{aligned} \qquad (1)$$

The objective function $f\colon R^n \to R$ and the equality constraint $c\colon R^n \to R^m$ with m<n are sufficiently smooth (i.e., their first derivatives must exist). These bounds are replaced by a logarithmic barrier term, which is added to the objective term to give:

$$\begin{aligned} \min \quad & \varphi_\mu(x) = f(x) - \mu \sum_i \log(x^{(i)}) \\ s.t \quad & c(x) = 0 \end{aligned} \qquad (2)$$

The barrier method solves a sequence of barrier problems for a decreasing μ_ℓ of barrier parameters with $\lim_{l\to\infty} \mu_l = 0$ to increasing tighter tolerance ε_ℓ with $\lim_{l\to\infty} \varepsilon_l = 0$. Under certain assumption it can be shown that a sequence of $x_*(\mu_\ell) = 0$ of Eqn. (2) (approximate) local solutions converges to a local solution of the original NLP of Eqn. (1) (Fiacco and McCormick, 1990). Since the exact solution $x_*(\mu_\ell) = 0$ is not of interest for large values of μ_ℓ, the corresponding barrier problem is solved only to a relaxed accuracy ε_ℓ with $\lim_{l\to\infty} \varepsilon_l = 0$.

The NLP solvers LOQO and KNITRO both implement Interior Point methods (also known as barrier method) for solving nonlinearly constrained optimization problems. The small differences in these solvers lead to performance differences exhibited. LOQO uses line-search merit functions whereas KNITRO uses trust-region merit functions to promote convergence. The NLP solver SNOPT implements a Sequential Quadratic Programming (SQP) method for solving large nonlinearly constrained optimization problems whereas the NLP solver LANCELOT implements a trust-region minimization of bound constrained augmented Lagrangian functions using Newton's Method. Lastly the NLP solver MINOS implements a reduced-gradient method with quasi-Newton approximations to the reduced Hessian for linearly constrained problems. It also employs a sequential linearly constrained (SLC) algorithm derived from Robinson's method for nonlinear constraints to solve the NLP. A classification of these NLP solvers is shown in Fig. 1.

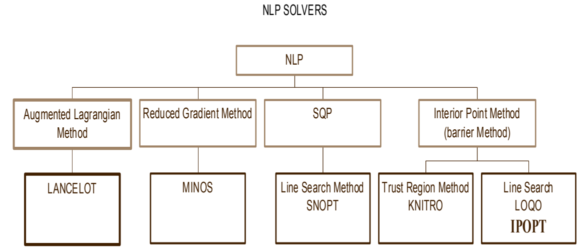

Figure 1. Summary of NLP solvers

General Representation of a Blending Model

The classical blending problems (Audet et al., 2000) are usually formulated as a linear program whereas the pooling problems have nonlinear terms and may be formulated as a bilinear program (BLP) as represented below:

$$
\begin{aligned}
\max \quad & \text{profit} = \sum_{t} \Big(\sum_{k} c_k f_{t,k} - \sum_{i} c_i f_{t,i} \Big) \\
s.t. \quad & \sum f_{t,jk} - \sum f_{t,ij} + v_{t+1,j} = v_{t,j} \\
& f_{t,k} - \sum f_{t,jk} = 0 \\
& \sum q_{t,j} f_{t,jk} - \sum q_{t,i} f_{t,ij} = 0 \\
& q_{k_{\min}} \le q_{t,k} \le q_{k_{\max}} \\
& v_{j_{\min}} \le v_{t,j} \le v_{j_{\max}}
\end{aligned}
$$

where indices i, j, k and t refer to crude, intermediate, products and time, respectively, and the variables f, q, and v are flows, tank qualities and tank volumes, respectively. The objective is derived through the input of the source pools and the output of the final pools. Since the qualities blend nonlinearly, bilinear terms are introduced in the model and with that the computational time increases. Moreover, the qualities themselves are often nonlinear functions of the flow rates.

There were 3 categories of blending models that were formulated in this study. The first category consists of 3 simple (1-day) models (Haverly, 1978; Audet and Hansen, 1998; Audet et al., 2000) with the measure of difficulty seen in the increase in the number of blending tanks to the product tanks. The second category consists of extending the 3 simple (1-day) models to run on a multiperiod basis (100-days) and the third category applies the bilinear programming formulation on an industrial problem to run on a 10-day cycle.

Numerical Results

Results from IPOPT solver were tested on a Dual Pentium 800MHz running Linux. Results from the others solvers were obtained from the NEOS solver's website. An initialization strategy was developed to reduce computational time.

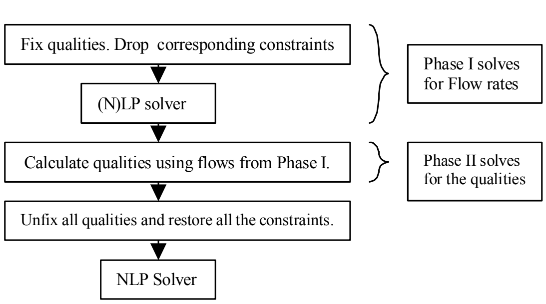

Figure 2. Initialization Strategy

Let: Haverly Model =HM
Audet & Hansen Model =AHM
Rehfeldt & Tisljar Model =RTM
Honeywell Model = IHM
number of variables =N
number of constraints =M
number of superbasic variables =S

Table 1. Results from Category I

HM Day 1 N=15,M=10,S=3	# of iter	Obj.	CPU (sec)
LANCELOT	10	3200	0.03
MINOS	3	3200	0.02
SNOPT	4	3200	0
KNITRO	15	3200	0
LOQO	22	3200	0.001
IPOPT	26	3200	0
AHM Day 1 N=20,M=13,S=3	# of iter	Obj.	CPU (sec)
LANCELOT	8	576	0.03
MINOS	3	576	0.02
SNOPT	4	576	0.02
KNITRO	19	576	0.05
LOQO	22	576	0.08
IPOPT	22	576	0.01
RTM Day 1 N=46,M=35,S=17	# of iter	Obj.	CPU (sec)
LANCELOT	10	3596	0.05
MINOS	17	3596	0.02
SNOPT	21	3596	0.04
KNITRO	*	*	*
LOQO	24	3596	0.12
IPOPT	17	3596	0.01

* indicates maximum iteration exceeded

Table 2. Results from Category II

HM Day 100 N=1500,M=1000,S=300	# of iter	Obj.	CPU (sec)
LANCELOT	11	3200	0.57
MINOS	381	3200	4.98
SNOPT	134	3200	0.46
KNITRO	18	3200	11.42
LOQO	26	3200	1.55
IPOPT	28	3200	1.78
AHM Day 100 N=2000,M=1300,S=300	# of iter	Obj.	CPU (sec)
LANCELOT	7	576	0.59
MINOS	318	576	4.09
SNOPT	108	576	0.51
KNITRO	20	576	19.24
LOQO	27	576	1.97
IPOPT	30	576	4.00

RTM Day 100 N=4600,M=3500,S=1700	# of iter	Obj.	CPU (sec)
LANCELOT	70	3596	1444.57
MINOS	2209	3596	34.07
SNOPT	1932	3596	6.07
KNITRO	**	**	**
LOQO	134	3596	72.82
IPOPT	29	3596	15.21

** indicates solver failure

Table 3. Results from Category III

IHM Day 1 N=1985, M=1585, S=1449	# of iter.	Obj.	CPU (sec)
LANCELOT	388	61.35	11736.26
MINOS	2274	61.35	3.5
SNOPT	**	**	**
KNITRO	37	100.3	157.87
LOQO	***	***	***
IPOPT	25	61.35	3.05
IHM Day 10 N=20826, M=16074 S=15206	# of iter.	Obj.	CPU (sec)
LANCELOT	****	****	****
MINOS	**	**	**
SNOPT	**	**	**
KNITRO	**	**	**
LOQO	***	***	***
IPOPT	65	26388	11064.44

**indicates solver failure; ***indicates primal/dual infeasible; ****indicates failure due to insufficient memory allocation

Conclusions

In category I, the results obtained were runs performed on 3 simple blending problems for a 1-day period. These simple models were extended to run for 100-days in category II. Comparing the performance of IPOPT with the other NLP solvers, there was significant improvement in computational speed as seen in the above results. In extending the model formulation to the industrial problem (Honeywell blending problem) in category III, there is improvement in computational speed when model is run for 1-day. Extending the model to run on a 10-day period, we note that a large increase in computational time is observed; this results because some of the gradient constraints become linearly dependent and cause the KKT matrix to become singular. Stabilized pivoting is implemented in IPOPT to treat this singularity. In addition, a preprocessing unit, which is still under development, will be used to handle the degeneracy in the model.

Future work will include incorporating component complementarity based models for discrete decision making in blending operations and this will include switching among the tanks as well as conditional relations in time. Global optimization of the blending models will also be under study.

Acknowledgements

Support from Honeywell International and the Chemical Engineering Department at Carnegie Mellon University is gratefully acknowledged for this research. We also thank Arvind Raghunathan for help with the IPOPT/AMPL interface.

References

Adhya, N., and N. Sahinidis (1999). A Lagrangian Approach to the Pooling Problem. *Ind. Eng Chem. Res,* **38**, 1956-1972.

Audet, C., and P. Hansen (1998). The Pooling Problem: Bilinear Programming Model. *Working Paper*.

Audet, C., P. Hansen, and J. Brimberg (2000). Pooling Problem: Alternate Formulations and Solutions Methods. *Les Cahiers du GERAD, G,* 2000-23.

Conn, A. R., I. Gould, and P. Toint (1992). *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization(Release 2), Springer Series in Computational Mathmatics,*Springer Verlag, Heidelberg, Berlin, **17**.

Fiacco, A., and G. P. McCormick (1990). *Nonlinear Programming Sequential Unconstrained Minimization Technique*, John Wiley, New York.

Fourer, R., D.Gay and B. Kernighan (1993). *AMPL. A Modeling Language for Mathematical Programming,*The Scientific Press Series, AT&T Bell Labs, pg. xiv.

Gill, P., W. Murray, and M. Wright (1981). P*ractical Optimization*, Academic Press, New York.

Gill, P., W. Murray, and M. Saunders (1997). *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,* Report NA , 97-2.

Haverly, C. A. (1978). *Studies of the behavior of Recursion for the Pooling Problem,* SIGMAP Bull*,* **25**

Murtagh, A. B., and M. Saunders (1993). *MINOS 5.0 users guide.* Technical Report SOL 83.20.

Nocedal, J., R. Bryd and R. Waltz (2000) Feasible Interior Point Methods Using Slacks for Nonlinear Optimization. Technical Report OTC, Optimization Technology Center. Northwestern University.

Vanderbei, J. R., and D. Shannon (1999). An Interior Point method for Nonconvex Nonlinear Programming. *Comp. Opt. Applics.,* **13**, 231-252.

Waechter, A., (2002). PhD Thesis, Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA.