2º CONGRESSO BRASILEIRO DE
P&D EM PETRÓLEO & GÁS

Applying Dynamic Programming to a Gas-Lift Optimization Problem

Eduardo Camponogara and Paulo H. R. Nakashima

Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Caixa Postal 476, 88040-900 Florianópolis, SC
e-mails: camponog@das.ufsc.br, phrn@das.ufsc.br

Abstract – The ever-increasing demand for nonrenewable resources and the pressure from stockholders are two forces pressing the oil industry for higher efficiency. The opportunities for advances abound in all sectors of the industry, in particular production processes in gas-lift oil wells, which are often favored to draw oil from high-depth reservoirs. Of concern in this paper is the task of distributing the limited supply of gas to the wells so as to induce an optimal oil production. Narrowing this task to the steady-state response of the wells gives rise to the gas-lift optimization problem, whose variables decide which wells should be active as well as the gas-injection and whose objective is profit maximization. The paper elaborates on a few properties of the problem and delivers a dynamic-programming algorithm to find approximate solutions. The effectiveness of the algorithm was demonstrated by contrasting its solutions against upper bounds obtained with continuous relaxation. As closure, the paper outlines a few directions for future research.

Keywords: gas-lifted oil production, optimization, dynamic programming.

1. Motivation

Reinforced by the pressure from competitive markets, the increasing demand for nonrenewable resources is compelling the oil industry to implement processes of higher efficiency and lesser environmental impact. The development of improved technologies is therefore a must to respond to these compelling forces, in all sectors of the industry. Of concern in this paper is the gas-lift operation of oil wells, not infrequently used to draw oil from high-depth reservoirs. In part due to the complexity of the processes thereof, there is room for improvement in a number of fronts, ranging from the modeling of the dynamic and steady-state behavior of the oil flow, through production optimization, to the automatic control of the processes (Wang et al., 2002).

Herein, the focus is on the optimization of the gas-lift operation of well clusters, taking into account the limits on gas supply, economic factors, and the nonlinear nature of the outflow, but confining the attention to its steady-state response. Roughly stated, the problem amounts to, first, deciding among a cluster of wells which will produce and, second, determining the gas-injection level for the active wells so as to induce maximum, overall efficiency. This problem, hereafter denoted by the gas-lift optimization problem, will be instantiated and solved continuously over time, often after the models of the well outflows are recalibrated and as the economic factors oscillate. The gas-lift optimization problem (GOP) consists of a mixed-integer nonlinear optimization problem, whose discrete variables indicate which wells are active and which are not, whose continuous variables gauge the amount of gas injected into the active wells, and whose constraints spell out the physical constraints (Plucenio, 2002; Grothey and McKinnon, 2002).

After introducing a mathematical formulation of the gas-lift optimization problem, the paper analyzes its computational hardness and delivers a dynamic-programming algorithm to find near-optimal solutions. Besides some analytical properties, we report results from numerical experiments with small, nevertheless representative problem instances. At the end, we elaborate on directions for future research and summarize the main contributions of the paper.

2. The Gas-Lift Optimization Problem

For a cluster of N wells, the gas-lift optimization problem can be cast in mathematical programming as follows:

$$\text{GOP:} \quad J = \text{Maximize } f = \sum_{n=1}^{N} [p_o\gamma_o^n q_o^n + p_g\gamma_g^n q_o^n - p_w\gamma_w^n q_o^n - p_i q_i^n] \tag{1.1}$$

$$\text{Subject to:} \quad q_o^n = \alpha_0^n y_n + \alpha_1^n q_i^n + \alpha_2^n (q_i^n)^2 + \alpha_3^n (q_i^n)^3 \qquad n = 1, \ldots, N \tag{1.2}$$

$$l_n y_n \le q_i^n \le u_n y_n \qquad n = 1, \ldots, N \tag{1.3}$$

$$\sum_{n=1}^{N} q_i^n \le q_{max} \tag{1.4}$$

$$y_n \in \{0, 1\} \qquad n = 1, \ldots, N \tag{1.5}$$

where:

- y_n takes on value 1 if the n-th well is active, otherwise it assumes value 0;
- q_i^n is the rate at which gas is injected into the n-th well, whose amount should lie between l_n and u_n;
- q_{max} is the upper bound on the overall gas injection, a physical constraint arising from the limited capacity of compression and gas storage;
- q_o^n is the flow of fluid from the n-th well, approximated as a polynomial function of the gas input flow;
- p_o and p_g are the prevailing market prices for oil and gas;
- p_w is the cost of fluid treatment before discharge;
- p_i is the cost incurred to compress the input gas; and
- γ_o^n, γ_g^n, and γ_w^n are respectively the fractions of oil, gas, and water that makeup the output flow at the n-th well, where $\gamma_o^n + \gamma_g^n + \gamma_w^n = 1$.

Assumption 1. *The output rate q_o^n is a concave function of the input rate q_i^n and $q_o^n \ge 0$ within the region $q_i^n \in [l_n, u_n]$.*

Assumption 2. *$p_o\gamma_o^n + p_g\gamma_g^n > p_w\gamma_w^n$ for each n.*

Remark 1. *f is a concave function of $q_i = [q_i^1, \ldots, q_i^N]$ inside the feasible region of GOP.*

Proof. For each n, $\beta_n = p_o\gamma_o^n + p_g\gamma_g^n - p_w\gamma_w^n$ is a positive constant and, therefore, $f_n = \beta_n q_o^n - p_i q_i^n$ is concave. Thus $f = f_1 + \ldots + f_n$ is a concave function. ■

The assumptions reflect what is typically observed, or can be satisfactorily approximated, in practice: i) the outflow increases as the gas-injection rises until a saturation point is reached, dwindling as the injection surpasses saturation, which is captured by the concavity of q_o^n; ii) the percentage of oil and gas production, γ_o and γ_g, should be greater than that of water for the well to be economically viable.

Proposition 1. *GOP is computationally hard.*

Proof. It is straightforward to reduce an instance of the knapsack problem (KP) (Wolsey, 1998) to an instance of GOP. Let I_{KP} be an instance of KP, consisting of a set $C = \{c_n : n = 1, \ldots, N\}$ with values of the objects, a set $W = \{w_n\}$ with their weights, and the capacity b of the knapsack. Generate an instance I_{GOP} as follows: set $q_{max} = b$; set $l_n = u_n = w_n$ for each n; set $\alpha_0{}^n = c_n$ and $\alpha_1{}^n = \alpha_2{}^n = \alpha_3{}^n = 0$ for each n; set $\gamma_o{}^n = 1$ and $\gamma_g{}^n = \gamma_w{}^n = 0$; and define $p_o = 1$ while $p_g = p_w = p_i = 0$. Clearly, an optimal solution $z = (y, q_i, q_o)$ to I_{GOP} induces an optimal solution to I_{KP}, namely the n-th item goes into the sack if and only if $y_n = 1$. ■

3. A Dynamic Programming Algorithm

Within the stochastic domain, dynamic programming (DP) (Bertsekas, 1995a) is concerned with scenarios where the decisions are made in stages, but whose outcomes can only be predicted to some extent, not anticipated. Within the deterministic domain, DP (Skiena, 1998) can be viewed as a technique for breaking up a problem into a series of smaller, easier to solve sub-problems that are coupled only to their preceding sub-problems. DP has been shown to be effective in a host of problems, in particular the knapsack problem (Wolsey, 1998), which is somewhat related to GOP. In what follows, we provide a dynamic-programming algorithm to approximately solve GOPs, along with theoretical insights into the quality of the approximations and ways of expediting the computations.

3.1. Recurrences

Restricting GOP to the subset $S_n = \{n, n + 1, \ldots, N\}$ of wells and the gas-compressing capacity q, rather than q_{max}, gives rise to the restricted gas-lift optimization problem $GOP_n(q)$. More formally, this problem can be stated as:

$$GOP_n(q): \quad J_n(q) = \text{Maximize} \sum_{k=n}^{N} [p_o\gamma_o{}^k q_o{}^k + p_g\gamma_g{}^k q_o{}^k - p_w\gamma_w{}^k q_o{}^k - p_i q_i{}^k] \tag{2.1}$$

$$\text{Subject to: } q_o{}^k = \alpha_0{}^k y_k + \alpha_1{}^k q_i{}^k + \alpha_2{}^k (q_i{}^k)^2 + \alpha_3{}^k (q_i{}^k)^3 \quad k = n, \ldots, N \tag{2.2}$$

$$l_k y_k \leq q_i{}^k \leq \min\{u_k, q\} y_k \quad k = n, \ldots, N \tag{2.3}$$

$$\sum_{k=n}^{N} q_i{}^k \leq q \tag{2.4}$$

$$y_k \in \{0, 1\} \quad k = n, \ldots, N \tag{2.5}$$

By noticing that $GOP \approx GOP_1(q_{max})$ and denoting $GOP_n = \{GOP_n(q) : 0 \leq q \leq q_{max}\}$, one could use the solution to GOP_n to solve GOP_{n-1}. This is precisely how DP iterates: it solves the series $\{GOP_n : n = 1, \ldots, N\}$ of problems, beginning with $n = N$ and progressing towards $n = 1$. In light of this fact, $GOP_n(q)$ can be recast recursively as follows:

$$P_n(q): \quad F_n(q) = \text{Maximize } f_n = p_o\gamma_o{}^n q_o{}^n + p_g\gamma_g{}^n q_o{}^n - p_w\gamma_w{}^n q_o{}^n - p_i q_i{}^n \tag{3.1}$$

$$\text{Subject to: } q_o{}^n = \alpha_0{}^n y_n + \alpha_1{}^n q_i{}^n + \alpha_2{}^n (q_i{}^n)^2 + \alpha_2{}^n (q_i{}^n)^3 \tag{3.2}$$

$$l_n y_n \leq q_i{}^n \leq \min\{u_n, q\} y_n \tag{3.3}$$

$$y_n \in \{0, 1\} \tag{3.4}$$

$$GOP_N(q): \quad J_N(q) = F_N(q) \tag{4}$$

$$GOP_n(q): \quad J_n(q) = \text{Maximize } (p_o\gamma_o{}^n q_o{}^n + p_g\gamma_g{}^n q_o{}^n - p_w\gamma_w{}^n q_o{}^n - p_i q_i{}^n) + J_{n+1}(q - q_i{}^n) \tag{5.1}$$

$$\text{Subject to: } q_o{}^n = \alpha_0{}^n y_n + \alpha_1{}^n q_i{}^n + \alpha_2{}^n (q_i{}^n)^2 + \alpha_2{}^n (q_i{}^n)^3 \tag{5.2}$$

$$l_n y_n \leq q_i{}^n \leq \min\{u_n, q\} y_n \tag{5.3}$$

$$y_n \in \{0, 1\} \tag{5.4}$$

Remark 2. *$J_n(q)$ is a nondecreasing function of q for $n = 1, \ldots, N$.*

Proof. After eliminating $q_o{}^k$, $k = n, \ldots, N$, using (2.2), the feasible region of $GOP_n(q)$ becomes $S_n(q) = \{z = (q_i{}^n, q_i{}^{n+1}, \ldots, q_i{}^N, y_n, y_{n+1}, \ldots, y_N) \in R^{N-n+1} \times \{0, 1\}^{N-n+1}$: subject to (2.3) and (2.4)$\}$. Whenever $q_a > q_b$, $S_n(q_a) \supseteq S_n(q_b)$ and therefore $J_n(q_a) \geq J_n(q_b)$. We conclude that $J_n(q)$ is nondecreasing in q. ■

To simplify further developments and the specification of the DP algorithm, we introduce the following notation:

- $R_n(q)$ denotes the set of values $q_i{}^n$ at which f_n attains its maximum, i.e., $R_n(q) = \{q_i{}^n : q_i{}^n \text{ solves } P_n(q)\}$;

- $r_n(q) = \text{Min}\{q_i^n \in R_n(q)\}$ is the smallest of the optimal solutions to $P_n(q)$; and
- $z_n(q)$ is an unconstrained maximum of f_n.

Remark 3. *The optimal solution $r_n(q)$ belongs to the set* $Q_n(q) = \{0, l_n, \min(q, u_n)\} \cup \{z_n(q) : l_n \leq z_n(q) \leq \min(q, u_n)\}$.

Proof. There are two possibilities: (*case 1*) if $y_n = 0$, then $q_i^n = 0$ and therefore $0 \in Q_n(q)$; (*case 2*) if $y_n = 1$, then due to the concavity of f_n and the convexity of the feasible set, $l_n \leq q_i^n \leq \min(q, u_n)$, it follows from the first-order optimality conditions that f_n attains its maximum either at the extremes or at the unconstrained maximum $z_n(q)$, if this is feasible. Hence, $Q_n(q) = \{0, l_n, \min(q, u_n)\} \cup \{z_n(q) : l_n \leq z_n(q) \leq \min(q, u_n)\}$. ∎

The Dynamic Programming Algorithm

Initialization.
For all $q \in [0, q_{max}]$ do
Compute $J_N(q)$
Recursion:
For $n = N - 1$ downto 1 do
For all $q \in [0, q_{max}]$
Compute $J_n(q) = \max\{F_n(r_n(h)) + J_{n+1}(q - r_n(h)) : 0 \leq h \leq q\}$

In its form, the above dynamic-programming algorithm entails solving an infinite number of problems, which would be practical only if there existed a differentiable, closed-form solution to $J_n(q)$ such as in the DP solution of the linear, quadratic, regulation problems that appears in the optimal control literature (Bertsekas, 1995a). In the presence of constraints and discrete variables, the solutions are in general obtained numerically with iterative algorithms. Our next developments propose an alternative DP algorithm to solve GOP approximately, avoiding the solution of an intractable number of problems by forcing q to take on discrete values.

3.2. Approximate Solution

By constraining q to assume values from a finite set $Q = \{q_0, \ldots, q_m\}$, one gains the ability to solve a tractable number of problems at the expense of solution quality. Hereafter we will assume that the discretization set Q is admissible, meaning that $q_0 = 0$, $q_m = q_{max}$, and $q_n \leq q_{n+1}$ for $n = 0, \ldots, m - 1$. The DP algorithm will fill out two tables: $Q[n, q]$ with the optimal solution to $\text{GOP}_n(q)$, and $J[n, q]$ with the objective values induced by these solutions, where $q \in Q$. The discrete DP algorithm is stated below.

The Discrete Dynamic Programming Algorithm

```
Initialization.
    For all q ∈ Q do
        Solve P_n(q), setting J[N, q] = F_n(q) and Q[N, q] = r_N(q)
    End-for
Recursion:
    For n = N – 1 downto 1 do  /* Compute J[n, q] using J[n + 1, q]  */
        For all q ∈ Q do
            J[n, q] = –∞
            Q[n, q] = 0
            For all q' ∈ Q, such that q' ≤ q do
                J' = J[n + 1, q'] + F_n(q – q')
                If J' > J[n, q] then
                    J[n, q] = J'
                    Q[n, q] = r_n(q – q')
                End-if
            End-for
        End-for
    End-for
```

Because $\text{P}_n(q)$ can be solved in time $\Theta(1)$ as shown in Remark 3, the discrete DP algorithm runs in time $\Theta(Nm^2)$ and its memory usage is of the order $\Theta(Nm)$, which is necessary to store tables $J[n, q]$ and $Q[n, q]$. In light of the fact that $J_n(q)$ is a nondecreasing function, according to Remark 2, we speculate that the running-time of the algorithm might be shortened to $O(Nm\log(m))$, by replacing its inner-loop with a search procedure akin to binary search (Bertsekas, 1995b; Cormen et al., 1990).

At termination, the discrete DP algorithm yields approximate solutions for a family of problems, namely $\{\text{GOP}_n(q) : q \in Q, n = 1, \ldots, N\}$, which encompasses an approximation to the original problem $\text{GOP} \approx \text{GOP}_1(q_{max})$. The profit predicted by the approximate solution to $\text{GOP}_n(q)$ is precisely the value $J[n, q]$—the corresponding solution can be computed in time $\Theta(Nm)$ with a backward search in the tables. Thus, in the event of an unanticipated drop of the gas-compression power, an approximate solution can be readily obtained from the tables. The quality of the approximate solution is, of course, highly dependent on the size and granularity of the discrete values for q, the elements of Q. The issues thereof will be addressed numerically in the next section.

Remark 4. *Let Q_a and Q_b be two admissible, dicretization sets for the gas-compression capacity q_{max}. If $Q_a \subseteq Q_b$, then the quality of the approximation induced by Q_a is not superior to the quality induced by Q_b, i.e.,* $J_{Q_b}[1,q_{max}] \geq J_{Q_a}[1,q_{max}]$.

Proof. Using induction in n, we can show that $J_{Q_b}[n, q] \geq J_{Q_a}[n, q]$ for all $q \in Q_a$. *Induction Basis, $n = N$*: $J_{Q_b}[N, q] = J_{Q_a}[N, q]$ for all $q \in Q_a$. *Induction Step, $n < N$*: $J_{Q_b}[n, q] = \text{Max}\{J_{Q_b}[n + 1, q'] + F_n(q - q') : q' \in Q_b\} \geq \text{Max}\{J_{Q_b}[n + 1, q'] + F_n(q - q') : q' \in Q_a\} \geq \text{Max}\{J_{Q_a}[n + 1, q'] + F_n(q - q') : q' \in Q_a\} = J_{Q_a}[n, q]$. Hence, $J_{Q_b}[1, q_{max}] \geq J_{Q_a}[1, q_{max}]$. ■

4. Numerical Experiments

This section gives an account of a number of numerical experiments, which were run to assess the quality of the solution produced by the discrete DP algorithm. The experimental set-up consists of two scenarios, one with a cluster of six wells and the other with twelve wells. Some of the parameters that make up the instances are depicted in Table 1. The parameters that are held constant across experiments are:

- the market price of oil, $p_o = 1.0$;
- the market price of gas, $p_g = 0.6$;
- the cost of gas compression and injection, $p_i = 0.1$; and
- the cost of water treatment before discharge, $p_w = 0.01$;

The lower and upper bounds on gas-injection are the same for all of the wells, that is, $l_n = l$ and $u_n = u$ for all n and some $l < u$, but these bounds will vary across experiments. The degree of discretization, the parameter m defining the cardinality of the discretization set Q, will be made to vary from one experiment to the next: more specifically, $Q = \{q_0, \ldots, q_m\}$ will be such that $q_0 = 0$, $q_m = q_{max}$, and $q_j - q_{j-1} = \delta$ for $j = 1, \ldots, m$, where δ is some suitable constant.

Table 2 shows the objective values of the solutions obtained with the DP algorithm as well as upper bounds J_{UB} for the optimal solutions, for a number of experiments that differ in: the number of wells, the maximum compressing-capacity q_{max}, and the maximum upper bound on gas-injection. These results provide evidence that the DP algorithm can yield near-optimal solutions, within a fraction of percentage from the upper bounds. (The upper bounds were obtained with continuous relaxation of the discrete variables, which renders a concave optimization problem for which efficient algorithms exist (Bertsekas, 1995b; Nocedal and Wright, 1999).)

5. Future Research Directions

Albeit progress has been made towards automating the decision-making, optimization, and control processes of gas-lift oil production, there remains a number of fronts for further progress. In particular, we foresee fruitful research and development along the following directions:

- *Stochastic models of the dynamic and steady-state out-flow response to gas injection:* instead of representing the outflow as a single function of the gas-injection, the outflow could be modeled with a family of functions, as well as their respective probabilities, to counter the uncertainties of the process identification.
- *Logic constraints:* the scenario in which the operation of certain wells is contingent on others, perhaps for convenience, safety, or other technological reasons, could well be expressed in terms of logic constraints.
- *Gas pressure variation and loss:* as the gas travels in a pipeline its pressure declines, possibly to the extent that the pressure loss should be accounted for and incorporated in the model.
- *Large-scale problem instances:* the oil production can take place over a wide area and employ multiple gas-compression units, one located at the central platform and the others distributed over the well clusters; in such a scenario the gas flows from the central unit to the nearby units, which raises the issue of how gas should be distributed so as to induce an optimal, overall production.

Incidentally, we have achieved some preliminary, nevertheless firm results towards the introduction of logic constraints and the use of stochastic models—for the former, families of implications that are acyclic can be naturally handled by our DP algorithm; for the latter, expectation maximization and worst-case analysis seem promising to handle stochastic models, in part because they allow the use of standard optimization techniques.

Table 1. Well Parameters

Well (k)	α_0	α_1	α_2	α_3	γ_o	γ_g	γ_w
1	0	3.2210	1.6743	–0.1549	0.70	0.20	0.10
2	0	3.1210	1.5743	–0.1649	0.75	0.17	0.08
3	0	2.7210	1.7743	–0.1749	0.65	0.25	0.10
4	0	2.9210	1.4743	–0.1449	0.65	0.20	0.15
5	0	2.7210	1.6743	–0.1549	0.60	0.30	0.10
6	0	3.4210	1.5743	–0.1649	0.80	0.12	0.08
7	0	2.0000	1.0000	–0.1200	0.65	0.30	0.05
8	0	3.8000	1.8000	–0.2000	0.80	0.17	0.03
9	0	3.0000	2.0000	–0.2500	0.60	0.25	0.15
10	0	4.0000	1.0000	–0.2800	0.70	0.23	0.07
11	0	2.5000	2.0000	–0.2500	0.65	0.25	0.10
12	0	2.7000	1.7000	–0.2400	0.75	0.20	0.05

Table 2. Numerical Experiments

Wells	l	u	q_{max}	m	$J[1, q_{max}]$	Upper Bound (J_{UB})	$(J_{UB} - J[1,q_{max}])$ / J_{UB}
1-6	3.65	10	40	5	190.9578	215.8527	11.54%
1-6	3.65	10	40	30	215.1865	215.8527	0.31%
1-6	3.65	10	20	5	114.0185	119.1060	4.27%
1-6	3.65	10	20	30	118.9109	119.1060	0.16%
1-6	3.65	6	20	5	114.0185	119.1054	4.27%
1-6	3.65	6	20	25	119.0741	119.1054	0.03%
1-12	3.65	10	70	10	335.2706	356.9148	6.06%
1-12	3.65	10	70	50	351.1864	356.9148	1.61%
1-12	3.65	10	30	10	178.8082	181.4667	1.47%
1-12	3.65	10	30	50	181.2690	181.4667	0.11%
1-12	3.65	6	30	10	178.8082	181.4667	1.47%
1-12	3.65	6	30	50	181.2690	181.4667	0.11%
Average							2.62%

6. Summary

The work reported heretofore dealt with a gas-lift optimization problem (GOP) whose applications are found in the oil industry. The paper looked into a few facets of the problem and proposed a discrete dynamic-programming (DP) algorithm to approximately solve GOP. The DP algorithm produced near-optimal solutions in a number of experiments.

7. Acknowledgments

This research was funded in part by Agência Nacional de Petróleo, under grant PRH 34/aciPG/ANP, and by a grant from PRPG/FunPesquisa, Universidade Federal de Santa Catarina. We also thank Agustinho Plucênio for sharing his expertise in oil-production processes, which were invaluable to this work.

8. References

BERTSEKAS, D. P. *Dynamic programming and optimal control,* Athena Scientific, 1995a.

BERTSEKAS, D. P. *Nonlinear Programming*, Athena Scientific, 1995b.

CORMEN, T.H., LEISERSON, RIVEST, R.L. *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

GROTHEY, A., MCKINNON, K. Decomposing the optimization of a gas lifted oil well network. Technical report MS-00-005, Department of Mathematics and Statistics, University of Edinburgh, March, 2002.

NOCEDAL, J., WRIGHT, S. J. *Numerical Optimization*, Springer-Verlag, 1999.

PLUCENIO, A. Stabilization and optimization of an oil well network operating with continuous gas lift. In: *SPE Int. Student Paper Contest, SPE Annual Technical Conference and Exhibition,* October, 2002.

SKIENA, S. S. *The algorithm design manual*, Springer-Verlag, 1998.

WANG, P., LITVAK, M. L., AZIZ, K. Optimization of production from mature fields. In: *Strategies for World Energy: The Development of the Oil and Gas Industry,* September, 2002.

WOLSEY, L. A. *Integer programming,* John Wiley & Sons, 1998.