

## **An Algorithm to Solve a Gas Cash-Out Problem**

Vyacheslav Kalashnikov, (E-mail: [kalash@osos.fime.uanl.mx](mailto:kalash@osos.fime.uanl.mx)), Universidad Autónoma de Nuevo León, Mexico

Roger Z. Ríos-Mercado, (E-mail: [roger@valma.fime.uanl.mx](mailto:roger@valma.fime.uanl.mx)), Universidad Autónoma de Nuevo León, Mexico

### **Abstract**

*In this paper, an algorithm for solving a natural gas cash-out problem is proposed. The problem is modeled as a mixed bi-level linear programming problem. Our proposed algorithm is based on a direct iterative scheme which splits the computation into two almost independent stages: (1) search for the pipeline's optimal response and, based upon that, the gas shipper's optimal distribution of the last day gas imbalances; (2) verifying the feasibility of the obtained optimal distribution of the last day imbalances (which involves solution of a quadratic programming problem). This idea of splitting leads to a computational scheme that may be valid for a wide enough class of bilevel programming problem. The algorithm is illustrated by solving a test instance.*

### **Introduction**

The field of multilevel optimization today is a well-known and important research field. A particular case of the bi-level programming problem (*cf.* [1]) is presented by the following mixed-integer model arising from the problem of minimization of cash-out penalty cost of a natural gas shipping company. The problem in question arises when a shipper draws a contract with a pipeline company to distribute a certain amount of gas among several points. What is actually delivered may be more or less of the amount that had been originally agreed upon (this phenomenon is called *an imbalance*). When an imbalance occurs, the pipeline penalizes the shipper by imposing a cash-out penalty policy. As this penalty is a function of the operating daily imbalances, an important problem for the shippers is how to carry out their daily imbalances so as to minimize their incurred penalty. It is easy to show that a very simple strategy by the shipper is to increase his imbalance day after day, to reach the maximum positive final day total imbalance. However, this strategy leads to some extra pseudo-storage and therewith, to the lost profits, one needs to introduce a payment for the gas storage by the shipper, to prevent his extra pseudo-storage.

In [2], a mathematical framework for the problem described above is presented. The problem is modeled as a mixed-integer bi-level linear programming problem (BLP) in which the shipper is a leader (the upper level) and the pipeline acts as a follower (lower level). A penalty function method to solve the BLP is also provided in [2].

However, the algorithm presented in [2] does not take into account certain specific features of the problem in question. For instance, the problem's structure allows one to split the computation into two almost independent stages: (1) search for the pipeline's optimal response and, based upon that, the gas shipper's optimal distribution of the last day gas imbalances; (2) verifying the feasibility of thus obtained optimal distribution of the last day imbalances (which involves solution of a quadratic programming problem). This idea of splitting leads to a computational scheme that may be valid for a wide enough class of bilevel programming problem. Another feature of the new algorithm is transforming the mixed integer programming problem to a standard nonlinear programming one, to get rid of the inconveniences of the presence of integer variables.

In this paper, we include a description of the model, a presentation of the proposed scheme, a computational evaluation with a small instance, and future research directions.

## Problem Specification

We describe the problem in terms of [1], in order to illustrate the proposed methodology. Assume that a shipper has entered into a contract (with clients) to deliver a given amount of natural gas from a receipt to a delivery meter in a given time frame. The shipper must stipulate title transfer agreement with the meter operators and a transportation agreement with the pipeline. Under such agreements, the shipper nominates a daily amount of gas to be injected by the receipt meter operator into the pipeline and to be withdrawn by the delivery meter operator from the pipeline. The pipeline transports the gas from the receipt meter to the delivery meter. Shippers daily nominate gas flows taking into account the constraints deriving from their buy/sell activity, their contractual constraints, and future market opportunities. The gas price is one of the major factors affecting their decisions. In the absence of cash-out provisions, historically shippers would take out high cost gas in the winter from the pipeline (causing negative imbalances), and pay the transporter back with low cost gas in the summer. This corresponds to a speculative behavior by the shippers, whereby imbalances are created and managed as pseudo-storage in order to take advantage of movements in the gas price. In the framework below, shippers are concerned with minimizing the sum of the cash-out penalties and the pseudo-storage fees.

## Notation

As stated above, the decision making process for the shipper (leader) is to determine how to carry out its daily imbalances so as to minimize the sum of the penalty and the storage fee that will be imposed by the pipeline (follower). The following notation is used to describe the model.

### Indices and Sets

- $i, j, k$  - zone pool indices;  $i, j, k \in J = \{1, \dots, P\}$ ;  
 $t$  - time index;  $t \in T = \{1, \dots, N\}$ .

### Parameters

- $x_{ii}^L, x_{ii}^U$  - bounds on daily imbalances at (end of) day  $t$  in zone  $i$ ;  
 $x_t^L, x_t^U$  - bounds on total daily imbalances at (end of) day  $t$ ;  
 $s_{ii}^L, s_{ii}^U$  - bounds on balance swings during day  $t$  in zone  $i$ ;  
 $e_{ij}$  - percentage of fuel retained for moving one dekatherm (dt) of gas from zone  $i$  to zone  $j$ ,  $i < j$ ;  
 $f_{ij}$  - transportation charge for moving one dt of gas from zone  $i$  to zone  $j$ ,  $i < j$ ;  
 $b_{ij}$  - backward haul credit for moving one dt of gas from zone  $j$  to zone  $i$ ,  $i < j$ ;  
 $x_{0j}$  - initial imbalance (start of day 1) in zone  $j$ ;  
 $r_j$  - the shipper's cash-out revenue [penalty] for having 1 dt of the positive [negative] final imbalance in zone  $j$ ;  
 $\beta_j$  - parameter of the shipper's payment for the pseudo-storage of gas in zone  $j$ ;  
 $\epsilon > 0$  - the tolerance parameter for the stopping criterion.

### Decision Variables

- $x_{ii}$  - imbalance at (end of) day  $t$  in zone  $i$ ;  
 $s_{ii}$  - imbalance swing during day  $t$  in zone  $i$ ;  
 $y_i$  - final imbalance in zone  $i$ ;  
 $u_{ij}$  - forward haul volume moved from zone  $i$  to zone  $j$ ,  $i < j$ ;  
 $v_{ij}$  - backward haul volume moved from zone  $j$  to zone  $i$ ,  $i < j$ ;  
 $z$  - total cash-out revenue for shipper.

### Auxiliary variables

- $q$  - binary variables equal to 1 (0) if final imbalances are non-negative (non-positive). In the particular case when all the imbalances are zeros set  $q=1$ .

*Remark 1.* In this simple model, we allow to transport the gas from zone  $i$  to zone  $j$  if and only if  $i < j$ . In more general cases, one can define a subset  $K$  of all possible pairs  $(i, j)$  such that  $u_{ij}$  and  $v_{ij}$  may be positive if and only if  $(i, j) \in K$ .

**Mathematical Model**

Here we provide the set of constraints and the objective functions involved in both the upper and lower levels of the problems.

**Upper Level Model**

*Objective:* Shipper's revenue to be maximized:

$$\max h_1(x, s, y, u, v, z) = z, \tag{1}$$

*subject to:*

$$x_{ii}^L \leq x_{ii} \leq x_{ii}^U, \quad t \in T, i \in J; \tag{2}$$

$$s_{ii}^L \leq s_{ii} \leq s_{ii}^U, \quad t \in T, i \in J; \tag{3}$$

$$x_t^L \leq \sum_{i \in J} x_{ti} \leq x_t^U, \quad t \in T; \tag{4}$$

$$x_{ti} = x_{t-1,i} + s_{ti}, \quad t \in T, i \in J. \tag{5}$$

**Lower Level Model**

*Objective:* In many cases, both shipper and pipeline agree in a policy that represents a compromise between them two, so rather than minimizing revenue for shipper, it is agreed to minimize deviation from zero (and hence, minimizing the amount of cash transactions). Therefore, the following objective is pursued by the pipeline:

$$\min h_2(x, s, y, u, v, z) = |z|, \tag{6}$$

subject to the constraints below.

*Balance constraints:* These constraints identify the relationship between the imbalance at the final day  $N$ , forward and backward haul volumes, retained fuel, and final imbalance in zone  $j$ :

$$y_j = x_{N,j} + \sum_{i:i < j} (1 - e_{ij}) u_{ij} + \sum_{k:k > j} v_{jk} - \sum_{k:k > j} u_{jk} - \sum_{i:i < j} v_{ij}, \quad j \in J; \tag{7}$$

*Gas conservation:* This constraint ensures no gas loss occurs. Although it follows directly from (7) after summing up by all  $j$ , we keep it on to make the problem clearer to non-technical users.

$$\sum_{j \in J} y_j + \sum_{(j,k): j < k} e_{jk} u_{jk} = \sum_{j \in J} x_{N,j}. \tag{8}$$

Note that  $\sum_{(j,k)} e_{jk} u_{jk} \geq 0$ , hence  $\sum_j y_j \leq \sum_j x_{N,j}$ .

*Zone upper bounds:* These constraints prevent cyclic movements of gas. It simply states that, from any given zone, one cannot move more than the initial positive imbalance.

$$\sum_{k:k > j} u_{jk} + \sum_{i:i < j} v_{ij} \leq \max\{0, x_{N,j}\}, \quad j \in J. \tag{9}$$

*Forward haul upper bounds:* These bounds prevent positive-to-positive and negative forward movement of imbalances.

$$u_{ij} \leq \begin{cases} x_{N,i} & \text{if } x_{N,i} > 0 \text{ and } x_{N,j} < 0; \\ 0 & \text{otherwise.} \end{cases} \tag{10'}$$

In the GAMS program, the latter constrained is represented as the following upper bound

$$u_{ij} \leq \max\{0, \min\{x_{N,i}, (-x_{N,j})\}\}, \quad i, j \in J, i < j. \quad (10)$$

*Backward haul upper bounds:* These bounds prevent positive-to-positive and negative backward movement of imbalances.

$$v_{ij} \leq \begin{cases} x_{N,j} & \text{if } x_{N,j} > 0 \text{ and } x_{N,i} < 0; \\ 0 & \text{otherwise.} \end{cases} \quad (11')$$

In the GAMS program, the latter constrained is represented as the following upper bound

$$v_{ij} \leq \max\{0, \min\{(-x_{N,i}), x_{N,j}\}\}, \quad i, j \in J, i < j. \quad (11)$$

Note that (10) is tantamount to (10'), as well as (11) to (11'), due to constraint (9) and the bounds of final imbalance below.

*Bounds on final imbalances:* These bounds ensure that all final imbalances have the “right” sign, i.e. an imbalance must not change sign at the end of the final day.

$$\min\{0, x_{N,j}\} \leq y_j \leq \max\{0, x_{N,j}\}, \quad j \in J. \quad (12')$$

In the GAMS program, we model the latter constraint by the following equivalent condition:

$$y_j \cdot x_{N,j} \geq 0, \quad j \in J. \quad (12)$$

*Sign of final imbalances:* This is a business rule that states that final imbalances for all zones must have the same sign (i.e. all non-positive or all non-negative); that means that the imbalances must not change sign from zone to zone:

$$-M(1-q) \leq y_j \leq Mq, \quad j \in J, \quad (13')$$

where  $M$  is a large enough number and  $q$  is a binary 0-1 variable. In order to get rid of the binary variable, we replace (13) in the GAMS program by the equivalent constraint:

$$y_i \cdot y_j \geq 0, \quad i, j \in J. \quad (13)$$

*Shipper's revenue:* This equation represents the revenue from the shipper's point of view:

$$z = \sum_{j \in J} [r_j y_j - \beta_j (y_j)_+^2] + \sum_{(i,j): i < j} b_{ij} v_{ij} - \sum_{(i,j): i < j} f_{ij} (1 - e_{ij}) u_{ij}; \quad (14)$$

here  $(y_j)_+ = \max\{0, y_j\}$ , and  $\beta_j > 0, j \in J$ , are the penalty factors which determine the fine imposed by the pipeline on the shipper for the final positive imbalance at pool  $j$ .

*Variable types:*

$$y_j, z \text{ free}; \quad (15)$$

$$u_{ij}, v_{ij} \geq 0; \quad (16)$$

In the next section, we describe the techniques of solution of the bi-level problem (1)-(16).

### Algorithm

Repeating the motivation given in the Introduction, we are going to split the computational scheme into two almost independent stages. First, we find an approximation of the optimal last day imbalance distribution  $x_N$  for the gas shipper by a gradient or stochastic search procedure, solving the pipeline's optimal response problem at each step of the latter procedure. Then we check the feasibility of the approximate optimal vector  $x_N$  (or, the possibility to reach this last day distribution from a given first day imbalance distribution) by solving a special quadratic programming problem.

Now to describe the algorithm in detail, consider the resulting bi-level programming problem:

$$\max h_1(x,s,y,u,v,z)=z, \tag{17}$$

$$\text{subject to (2)—(5) and to:} \tag{18}$$

$$\min h_2(x,s,y,u,v,z)=z^2, \tag{19}$$

$$\text{subject to (7)—(16).} \tag{20}$$

We consider it as a bi-level hierarchical system in the Euclidean space where the upper level decision maker (hereafter the leader) controls the variable vector  $(x, s) \in R^{NP} \times R^{NP}$ , whereas the lower level (hereafter the follower) controls the vector of continuous variables  $w = (y, u, v) \in R^P \times R^{P(P-1)/2} \times R^{P(P-1)/2}$ . The leader makes his decision first by taking into account the reaction of the follower to his course of action. If we

introduce the function

$$F(y, u, v) = \sum_{j \in J} [r_j y_j - \beta_j (y_j)_+^2] + \sum_{(i,j):i < j} b_{ij} v_{ij} - \sum_{(i,j):i < j} f_{ij} (1 - e_{ij}) u_{ij}$$

and the subset  $W=W(x,s)$  as follows:

$$W=W(x,s)=\{w=(y,u,v) \text{ satisfying (7)—(13) and (15)—(16)}\},$$

then the reaction  $w^*=w^*(x,s)=(y^*,u^*,v^*)$  of the follower to the leader's decision  $(x,s)$  is a solution to the equilibrium problem provided by the following variational inequality:

$$\langle F(w^*) \nabla F(w^*), w - w^* \rangle \geq 0, \forall w \in W(x, s);$$

$$\text{here } \nabla F(y, u, v) = \left( (r_j - 2\beta_j (y_j)_+),_{j \in J}, (-f_{ij} (1 - e_{ij}))_{(i,j):i < j}, (b_{ij})_{(i,j):i < j} \right).$$

Now we are ready to describe the algorithm's steps.

**Data Input.** Input parameter values:

- (i)  $x_{0j}, j \in J = \{1, \dots, P\}$ ;
- (ii)  $x_{ij}^L, x_{ij}^U, s_{ij}^L, s_{ij}^U, x_t^L, x_t^U, j \in J, t \in T$ ;
- (iii)  $e_{ij}, b_{ij}, f_{ij}, i, j \in J, i < j$ ;
- (iv)  $r_j, \beta_j, j \in J$ ;
- (v)  $\varepsilon > 0$ .

**Preliminary Step.** Set  $k:=0$ , and solve the standard non-smooth quadratic programming problem (1)-(5),(7),(14) with  $u_{ij}=v_{ij}=0$  for all  $i,j$ . If the problem is solvable and thus obtained optimal vector  $x$  has the last day coordinates all non-positive or all non-negative, i.e. if  $x_{N,j} \geq 0$  or  $x_{N,j} \leq 0, j \in J$ , then this  $x$  is obviously an optimal solution of the initial problem as well. Otherwise, set  $x_N^{(k)} = (x_{N,1}^{(k)}, \dots, x_{N,P}^{(k)})^T := (x_{N,1}, \dots, x_{N,P})^T$ . GOTO Step 1.

**Step 1. Lower Level Problem.** Given  $x^{(k)}$ , solve the lower level problem

$$z^2 = F^2(y, u, v) \rightarrow \min_{u, v \geq 0}, \tag{21}$$

subject to (7)—(16). Denote the solution of (21) by  $u^{(k)} = u^{(k)}(x^{(k)})$ ,  $v^{(k)} = v^{(k)}(x^{(k)})$ ; the final imbalance vector  $y^{(k)}$  is then obtained by (7). GOTO Step 2.

**Step 2. Shift Evaluation.** After having calculated  $y^{(k)}$  by (7) as follows:

$$y_j^{(k)} = x_{N,j}^{(k)} + \sum_{i:i < j} (1 - e_{ij}) u_{ij}^{(k)} + \sum_{m:m > j} v_{jm}^{(k)} - \sum_{m:m > j} u_{jm}^{(k)} - \sum_{i:i < j} v_{ij}^{(k)}, \quad j \in J,$$

set

$$d^{(k)} := \nabla_y F(y^{(k)}, u^{(k)}, v^{(k)}) = [r_j - 2\beta_j (y_i^{(k)})_+]_{j \in J}. \tag{22}$$

If  $\|d^{(k)}\| < \varepsilon$  then GOTO Step 5, otherwise GOTO Step 3.

**Step 3. Upper Level Problem.** Find a candidate for the  $(k+1)$ -th iterate as follows:

$$\tilde{x}^{(k+1)} := x^{(k)} + \lambda_k d^{(k)},$$

where

$$\lambda_k \in \underset{\lambda \in R}{\text{Arg max}} \Phi^{(k)}(\lambda) \tag{23}$$

with

$$\Phi^{(k)}(\lambda) := F(y(\lambda), u(\lambda), v(\lambda)). \tag{24}$$

Here

$$y(\lambda) = y(x(\lambda), s(\lambda)),$$

$$u(\lambda) = u(x(\lambda), s(\lambda)),$$

$$v(\lambda) = v(x(\lambda), s(\lambda))$$

are solutions of problem (21) with

$$x(\lambda) := x^{(k)} + \lambda \xi d^{(k)},$$

where  $\xi$  is a pseudo-random variable with the uniform distribution over the interval  $[-1,1]$ , and  $s(\lambda)$  is any feasible vector of the auxiliary problem presented in Step 4 below. The purpose of this randomization is to widen the region of search, as the function (24) may not be convex in the general case.

**Step 4. Feasibility Test.** Solve the QP problem:

$$\sum_{t \in T} \sum_{j \in J} s_{ij}^2 \rightarrow \min, \tag{25}$$

subject to:

$$s_{ij}^L \leq s_{ij} \leq s_{ij}^U, \quad j \in J, t \in T; \tag{26a}$$

$$x_{1j}^L \leq x_{0j} + s_{1j} \leq x_{1j}^U, \quad j \in J; \tag{26b}$$

$$x_1^L \leq \sum_{j \in J} (x_{0j} + s_{1j}) \leq x_2^U; \tag{26c}$$

.....

$$x_{N-1,j}^L \leq x_{0j} + \sum_{t=1}^{N-1} s_{tj} \leq x_{N-1,j}^U, \quad j \in J; \tag{26d}$$

$$x_{N-1}^L \leq \sum_{j \in J} \left( x_{0j} + \sum_{t=1}^{N-1} s_{tj} \right) \leq x_{N-1}^U, \tag{26e}$$

$$x_{N,j}^L \leq x_{0j} + \sum_{t \in T} s_{tj} = x_j(\lambda) \leq x_{N,j}^U, \quad j \in J; \tag{26f}$$

$$x_N^L \leq \sum_{j \in J} \left( x_{0j} + \sum_{t \in T} s_{tj} \right) \leq x_N^U. \tag{26g}$$

Here (26f) is a key constraint: if the point  $x(\lambda) = x^{(k)} + \lambda \xi d^{(k)}$  satisfies it, we can solve the QP problem (25)–(26). In fact, we may solve it only until having discovered any feasible solution  $s(\lambda)$ . In particular, if problem (25)–(26) is feasible for  $\lambda = \lambda_k$ , then set  $x^{(k+1)} := \tilde{x}^{(k+1)}$ ,  $k := k+1$ , and GOTO Step 5.

**Step 5. Stopping Criterion.** If  $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$  then STOP; otherwise GOTO Step 1.

To implement thus described algorithm, a program in GAMS [3] was composed to test problems of small size with  $N \leq 12$  and  $T \leq 6$ . The algorithm performed well and found the approximate solutions with the tolerances  $\varepsilon = 10^{-3}$  and  $\varepsilon = 10^{-4}$  quite fast. While making the numerical experiments we noticed that:

- 1) The shipper's revenue function  $z$  defined by (14) with  $\beta_j = 0, j \in J$ , leads to the very simple strategy by the shipper at the upper level: he is to increase his imbalance day after day and try to maximize his (positive) last day imbalance. As this strategy leads to an extra pseudo-storage, we have changed the leader's revenue function by subtracting the (non-smooth) quadratic terms  $\beta_j (y_j)_+^2$  with  $\beta_j > 0, j \in J$ , which limit the shipper's extra pseudo-storage.

We tested the GAMS program with a small model of the gas cash out problem with 4 pools and 2 time intervals (i.e.,  $P = 4$  and  $N = 2$ ). The initial data are defined in Tables 1 to 7.

**Table 1: Initial data**

Pools $j$	Initial imbalance $x_0$	Penalty factors $r_j$	Penalty coefficients $\beta_j$
Pool 1	-10.0	10.0	0.1
Pool 2	-4.0	8.0	0.2
Pool 3	3.0	6.0	0.3
Pool 4	6.0	4.0	0.4

**Table 2: Lower and upper bounds of total imbalance**

Time periods $t$	Lower bound $x_t^L$	Upper bound $x_t^U$
Day 1	-11.0	1.0
Day 2	-17.0	7.0

**Table 3: Transportation costs between pools  $f_{ij}$**

	Pool 1	Pool 2	Pool 3	Pool 4
Pool 1		2.0	4.0	4.0
Pool 2			2.0	2.0
Pool 3				1.0
Pool 4				

**Table 4: Percentage of fuel retained when transported between pools  $e_{ij}$**

	Pool 1	Pool 2	Pool 3	Pool 4
Pool 1		0.1	0.2	0.2
Pool 2			0.1	0.1
Pool 3				0.05
Pool 4				

**Table 5: Bonus for shipper for backward haul  $b_{ij}$**

	Pool 1	Pool 2	Pool 3	Pool 4
Pool 1		4.0	2.0	2.0
Pool 2			4.0	4.0
Pool 3				2.0
Pool 4				

**Table 6: Lower limits of imbalance at pool  $j$  on day  $t$ , i.e.  $x_{ij}^L$**

	Pool 1	Pool 2	Pool 3	Pool 4
Day 1	-15.0	-9.0	0.0	1.0
Day 2	-20.0	-13.0	-5.0	-4.0

**Table 7: Upper limits of imbalance at pool  $j$  on day  $t$ , i.e.  $x_{ij}^U$**

	Pool 1	Pool 2	Pool 3	Pool 4
Day 1	-5.0	-1.0	6.0	10.0
Day 2	0.0	-5.0	10.0	15.0

Swing of imbalance lower and upper bounds are:  $s_{ij}^L = -3, s_{ij}^U = 3 \forall i, j \in J$ .

After the Preliminary step starting from the initial imbalance  $x_0 = (-10, -4, 3, 6)^T$ , the last day imbalance  $x_N = (-4, 2, 9, 5)^T$  with the shipper's objective function value  $z = 14.9$ . As the latter vector does not have all its entries of the same sign, we proceed with the steps of the algorithm. With the final tolerance  $\varepsilon = 0.001$ , Step 3 obtains the final approximate of the last day imbalance  $x_N = (-12.495, -5.809, 9.0, 9.789)^T$  with the shipper's objective function value  $z = 50.072$ . The pipeline's optimal response to this shipper's strategy is the final imbalance  $y = (0.0; 0.0; 2.22 \cdot 10^{-16}; 0.485)^T$  with the same pipeline's objective function value  $z = 50.072$ . All the forward hauls are zeros, whereas the backward hauls are:  $v_{13} = 9.0; v_{14} = 3.495; v_{24} = 5.809$ . As the last day imbalance  $x_N = (-12.495, -5.809, 9.0, 9.789)^T$  satisfies the feasibility test of Step 4, i.e., it is obtained from the initial imbalance  $x_0 = (-10, -4, 3, 6)^T$  by the daily swings indicated in Table 8 below, the gas cash-out problem is solved, the solution being presented in Table 9 below.

**Table 8: Optimal swings of imbalance at pool  $j$  on day  $t$ , i.e.  $s_{ij}$**

	Pool 1	Pool 2	Pool 3	Pool 4
Day 1	-1.148	-0.805	3.000	1.994
Day 2	-1.347	-1.004	3.000	1.795

**Table 9: Optimal imbalance at pool  $j$  on day  $t$ , i.e.  $x_{ij}$**

	Pool 1	Pool 2	Pool 3	Pool 4
Day 1	-11.148	-4.805	6.000	7.994
Day 2	-12.495	-5.809	9.000	9.789

### Conclusions and Directions of Future Research

In this paper, we use a mathematical framework for the problem of minimizing the cash-out penalties from the point of view of the natural gas shipper. The problem is modeled as a bi-level nonlinear programming problem. To solve it efficiently, we describe an algorithm which actually splits the initial problem into two almost independent subproblems. The solution set of the first subproblem consists of all feasible imbalances on the last day of the given time period. Each of these feasible solutions can be chosen as an initial iterate for the second subproblem, the bi-level leader-follower one.

In our further research step, we are going to modify the model introducing the fines imposed by the pipeline on the shipper for the daily negative imbalances as well. When considering such a more complex problem we will have to apply subtler methods to split the problem into two subproblems mentioned above.

One of the future targets will also be an application of the optimal or quasi-optimal step accuracy control realized at the algorithm's iterations. For instance, as Step 3 is rather complex by involving the feasibility test

(Step 4) and solving the lower level problem (Step 1) for each probed point  $x(\lambda)$ , one has to cut the iterative procedures before long. An optimal control strategy for the accuracy while solving the lower level subproblem was proposed in [2] for general bi-level processes. Now we are going to apply these general techniques to the gas cash-out problem.

**Acknowledgements.** The research of the first author is financially supported by the UANL-PAICYT (Grant No. CA562-01). The research of the second author is financially supported by the UANL-PAICYT (Grant No. CA555-01) and by the CONACYT-Mexico (Grant No. J33187-A).

## References

- [1] J. F. Bard. “*Practical Bilevel Programming: Algorithms and Applications*”. Kluwer, Dordrecht, Netherlands, 1998.
- [2] V. V. Kalashnikov and R. Z. Ríos-Mercado. A Penalty Function Approach to a Mixed-Bilevel Programming Problem. In C. Zozaya, M. Mejía, P. Noriega, and A. Sánchez (editors), *Proceedings of the 3rd International Meeting on Computer Science, Vol. II*, 1045-1054. Aguascalientes, Mexico, September 2001.
- [3] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman. “*GAMS. A User’s Guide*”. GAMS Development Corporation, Potomac, USA, 1998.
- [4] V. V. Kalashnikov and N. I. Kalashnikova. Optimal Control of Accuracy in Bilevel Iteration Processes. *Optimizacia*, 44(61):27-55, 1988 (in Russian).