EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

European Journal of Operational Research 120 (2000) 30–46

www.elsevier.com/locate/orms

Theory and Methodology

A bilevel programming approach to determining tax credits for biofuel production [1]

Jonathan F. Bard [a,*], John Plummer [b,2], Jean Claude Sourie [c,3]

[a] Graduate Program in Operations Research, Department of Mechanical Engineering, University of Texas, Austin, TX 78712-1063, USA

[b] Department of Management Science and Information Systems, College of Business Administration, University of Texas, Austin, TX 78712, USA

[c] Institut National de la Recherche Agronomique (INRA), Centre de Grignon, Departement d'Economic et de Sociologies Rurales, 78850 Thiverval-Grignon, France

Received 13 October 1997; accepted 13 October 1998

Abstract

This paper presents a bilevel programming formulation of a leader–follower game that can be used to help decision makers arrive at a rational policy for encouraging biofuel production. In the model, the government is the leader and would like to minimize the annual tax credits it allows the petro-chemical industry for producing biofuels. The crops grown for this purpose are on land now set aside and subsidized through a different support program. The agricultural sector is the follower. Its objective is to maximize profits by selecting the best mix of crops to grow as well as the percentage of land to set aside. Two solution algorithms are developed. The first involves a grid search over the tax credit variables corresponding to the two biofuels under consideration, ester and ethanol. Once these values are fixed, nonfood crop prices can be determined and the farm sector linear program solved. The second algorithm is based on an approximate nonlinear programming (NLP) formulation of the bilevel program. An "engineering" approach is taken where the discontinuities in the government's problem are ignored and the farm model is treated as a function that maps nonfood crop prices into allocation decisions. Results are given for an agricultural region in the northern part of France comprising 393 farms.

Keywords: Nonlinear bilevel programming; Government regulation; Subsidies; Grid search algorithm; Agriculture

1. Introduction

The high cost associated with subsidizing the agricultural sector coupled with a desire to reduce the amount of automobile emissions has led the French government in conjunction with the European Union (EU) to explore the possibility of encouraging the petro-chemical industry to

* Coresponding author. Tel.: +1-512-471-3076; fax: +1-512-471-8727; e-mail: jbard@mail.utexas.edu

[1] This work was supported by a grant from Institut National de la Recherche Agronomique and the Texas Higher Education Coordinating Board under the Advanced Research Program – ARP 003.

[2] E-mail: jcplummer@mail.utexas.edu

[3] E-mail: sourie@grignon.inra.fr

0377-2217/00/$ – see front matter
PII: S 0 3 7 7 - 2 2 1 7 (9 8) 0 0 3 7 3 - 7

produce biofuels from farm crops. Examples of crops that can be used for this purpose are wheat, corn, rapeseed and sunflower among others. The stumbling block to this policy is that industry's costs for producing fuels from hydrocarbon-based raw materials is significantly less than it is for producing biofuels. Without incentives in the form of tax credits, industry will not buy farm output for conversion.

The problem faced by the government is to determine the level of tax credits for each final product or biofuel that industry can produce while minimizing public outlays. A secondary objective is to realize some predefined level of land usage for nonfood crops. Industry is assumed to be neutral in this scenario and will produce any biofuel that is profitable. In the analysis, the agricultural sector is represented by a subset of farms in an agriculturally intensive region of France and is a profit maximizer. It will use the land available for nonfood crops only as long as the revenue generated from this activity exceeds the difference between the set-aside payments now received directly from the government and the maintenance costs incurred under the support program. If a farmer leaves a plot of land fallow and wishes to qualify for direct aid, he must plant a cover crop during the growing season to enrich the soil. Currently, 15% of the arable land (with some exceptions for very small farmers) must remain fallow and is thus eligible for this type of aid.

The conflict inherent in the problem is that the government wants to minimize its costs subject to a given level of land usage for nonfood crops while the agricultural sector wishes to maximize its profits subject to the technological constraints of production and certain agronomic constraints that are part of the regulatory program. A typical agronomic constraint is that no more than 47.6% of the arable land on each farm may be allocated to wheat production for both food and nonfood crops.

In general, agricultural constraints result from observations of a representative sample of farms. We can observe that a crop (wheat, for instance) should not exceed a maximum limit due to agronomic (technical) constraints, such as the need for diversification, labor requirements, and the use of pesticides. This limit is expressed as a percentage of available land. From the sample farm, a cumulative distribution curve of this percentage can be built which is then extrapolated to all the farms of the sample (in doing so, it is assumed that the region is homogeneous from an agronomic point of view). For example, a limit of 47.6% implies that on each farm in the region, wheat can vary between 0% and 47.6% of the available land. The quantity actually grown depends on the profitability of wheat relative to the farmer's other options. This approach is applied to all crops (or group of crops such as cereals and oilseeds) to obtain a system of constraints that defines, in part, the feasibility set of activities in the farm model.

The regulatory scenario can be viewed as a classic leader–follower game in which the former sets policy and the latter reacts, sometimes with unforeseen consequences (Anandalingam and Friesz, 1992). The purpose of this paper is to describe the modeling details along with the highlights of our solution algorithms. In the next section, we give a bilevel programming formulation of the resultant problem (Aiyoshi and Shimizu, 1981; Bard and Falk, 1982; Bialas and Karwan, 1984; Fortuny-Amat and McCarl, 1981) followed in Section 3 with several extensions. We then describe two algorithms developed for finding solutions to our particular application. The first is based on the idea of imposing a grid on the leader's variable set and solving the follower's problem for each point enumerated. The second is a more traditional nonlinear programming (NLP) approach that assumes a standard model in which all functions are once continuously differentiable. SQP (Fan et al., 1988) is used to find solutions. The results in either case are almost identical, as shown in Section 5. We conclude in Section 6 with a comparative discussion of the two approaches.

2. Mathematical model

The following notation is used to describe the subsidy model under investigation.

Units

ha hectare (10,000 square meters)
hl hectoliter (100 liters)
t metric tonne (1000 kilograms)
FF French francs

Indices and sets

c index for food crops; $c \in C = \{1,\ldots,n_c\}$ (c' is index for sugar beets)
d index for nonfood crops; $d \in D \subset C$
f index for farms; $f \in F = \{1,\ldots,n_f\}$
b index for biofuels; $b \in B = \{1,\ldots,n_b\}$
k index for agronomic constraints; $k \in K = \{1,\ldots,n_k\}$
$C(f)$ subset of food crops grown on farm f
$D(f)$ subset of nonfood crops grown on farm f
$D(b)$ subset of nonfood crops that can be used to make biofuel b
$L(b)$ subset of nonfood crops subject to capacity limitations in the production of biofuel b (ethanol only); $L(b) \subseteq D(b)$
$B(d)$ subset of biofuels that can be made from crop d
$G(f, k)$ set of food crops grown on farm f associated with agronomic constraint k
$H(f, k)$ set of nonfood crops grown on farm f associated with agronomic constraint k

Parameters

m_{cf} gross margin (income) for food crop c grown on farm f (FF/ha)
r_{df} yield of nonfood crop d grown on farm f (t/ha)
α_{db} factor for converting one tonne of nonfood crop d to biofuel b (hl/t)
β_{db} cost of converting one unit of nonfood crop d to biofuel b (FF/hl)
c_{df} production cost for nonfood crop d on farm f (FF/ha)
σ_f total arable land available on farm f (ha)
σ'_f land available on farm f for sugar beets (for sugar) (ha)
s_d subsidy paid to farmers for nonfood crop d (FF/ha)
π_b profit expected by industry for one unit of biofuel b (FF/hl)
v_b market price for biofuel b (FF/hl)
o_{db} market price of co-products associated with production of one unit of biofuel b from nonfood crop d (FF/hl)
w_f multiplier used to scale up arable land of farm f
L_b limitations on production of biofuel b from certain nonfood crops $L(b)$ (currently 3,000,000 hl/yr of ethanol from corn, wheat, and sugar beets only)
ϕ_k maximum fraction of land permitted for crops included in agronomic constraint k
δ_d indicator parameter equal to 1 if no subsidy is paid for nonfood crop d grown on land set aside; 0 otherwise
θ fraction of arable land that must be set aside but could be used for production of nonfood crops (currently, $\theta = 0.15$ set by the EU)
γ set-aside payment for fallow land (currently 1600 FF/ha)
ρ fraction of set-aside land targeted by government for nonfood crop production (currently 0.20)
η fraction of cake produced from one tonne of either rapeseed or sunflower (approximately 0.56 for both crops)
μ large constant

Decision variables

x_{cf} area allocated to food crop c on farm f (ha)
xn_{df} area allocated to nonfood crop d on farm f (ha)
xf_f area set aside on farm f (ha)
p_d price at farm gate paid by industry for nonfood crop d (FF/t)
τ_b government tax credit given to industry for biofuel b (FF/hl)

Government model (*leader*)

$$\min \sum_{b\in B}\sum_{f\in F}\sum_{d\in D(f)} \alpha_{db} r_{df} xn_{df} \tau_b - \gamma \sum_{f\in F}\sum_{d\in D(f)} \delta_d xn_{df} \tag{1.1}$$

subject to

$$\sum_{f\in F}\sum_{d\in D(f)} xn_{df} \geqslant \rho\theta\sum_{f\in F} w_f\sigma_f, \tag{1.2}$$

$$\sum_{f\in F}\sum_{d\in D(f)\cap L(b)} \alpha_{db}r_{df}xn_{df} \leqslant L_b \quad \forall b\in B. \tag{1.3}$$

(Industry model)

$$p_d \leqslant (\tau_b + v_b - \beta_{db} - \pi_b + o_{db})\alpha_{db} \quad \forall d\in D,\ b\in B, \tag{1.4}$$

$$p_d \geqslant 0, \quad \tau_b \geqslant 0 \quad \forall d\in D,\ b\in B. \tag{1.5}$$

Agricultural sector model (*follower*)

$$\max \sum_{f\in F}\sum_{c\in C(f)} m_{cf}x_{cf} + \sum_{f\in F}\sum_{d\in D(f)} (p_d r_{df} + s_d - c_{df})xn_{df} + \gamma\sum_{f\in F} xf_f \tag{1.6}$$

subject to

$$\sum_{c\in C(f)} x_{cf} + \sum_{d\in D(f)} xn_{df} + xf_f \leqslant w_f\sigma_f \quad \forall f\in F, \tag{1.7}$$

$$\sum_{d\in D(f)} xn_{df} + xf_f = \theta w_f\sigma_f \quad \forall f\in F, \tag{1.8}$$

$$x_{c'f} \leqslant w_f\sigma'_f \quad \forall f\in F, \tag{1.9}$$

$$\sum_{c\in C(f)} x_{cf} + \sum_{d\in D(f)} xn_{df} \leqslant \phi_k w_f\sigma_f \quad \forall f\in F,\ k\in K, \tag{1.10}$$

$$x_{cf} \geqslant 0, \quad xn_{df} \geqslant 0, \quad xf_f \geqslant 0, \quad \forall c\in C,\ d\in D,\ f\in F. \tag{1.11}$$

In the model, the government assumes the role of leader with the objective (1.1) of minimizing the total value of tax credits given to the petrochemical industry. The second term in (1.1) reflects the savings to the government for not having to pay a premium, γ, to farmer f who grows nonfood crop d on land that would ordinarily be set aside. Currently, this applies only to sugar beets. When farmers grow sugar beets and only sugar beets on land set aside, they lose the subsidy per hectare on that land. But if sugar beet farms produce wheat for ethanol or rapeseed for ester on land set aside, they keep the subsidy. This policy is dictated by Brussels in the larger context of the EU.

The arable land available to farmer f in the sample group is denoted by σ_f. To account for the bias in the farm sample and to improve the fidelity of the model, each value of σ_f is scaled up by a unique multiplier denoted by w_f. This value reflects the yield, soil fertility, climate, and surface area of the specific farm with respect to a predetermined norm. The summation on the right hand side of constraint (1.2) is an approximation of the total arable land available in the region as a function of these factors. The parameter θ, currently equal to 0.15, is the proportion of arable land that must be set aside. Its value is fixed by the EU and may vary between 0% and 15% depending on the market situation and the existence of surpluses (for example, for the next year it has been set at 10%). If the arable land is maintained properly while fallow, it is eligible for a direct government payment denoted by γ. Such agreements are common in most western countries.

As an alternative for the land set aside, the farmer is permitted to grow certain crops that can be converted to fuels by the petro-chemical industry. The requirement in (1.2) that at least $\rho \times 100\%$ of the fallow land be used for the purpose of growing nonfood crops forces the government to grant tax credits since the cost to industry for producing fuels from biomass is currently greater than the cost of producing fuels from hydrocarbon sources. Note that ρ is a policy variable and is treated as a parameter in the model. Without incentives, industry would not buy any farm output for conversion. The price industry is willing to pay at the farm gate for nonfood crop d, denoted by p_d, must be high enough to offset the direct payment, γ, when subsidy payments, s_d, and costs, c_{df}, are taken into account. The tradeoff at the farm level is reflected in the follower's objective function (1.6) which will be discussed presently.

On the set-aside land, the agricultural sector can grow a variety of nonfood crops for sale to industry for biofuel production. Current installed capacity limits output in certain instances. Constraint (1.3) restricts the production of biofuel b to L_b hectoliters per year.

In the model, we assume that industry is a neutral player whose only objective is to make a profit. Constraint (1.4) assures that if crop d is purchased by industry to produce biofuel b, the tax credit τ_b is sufficiently high to cover costs β_{db} given the market price v_b, an expected profit π_b, as well as any revenues from co-products o_{db}. The neutrality assumption implies that the government will control the choices of p_d. A more elaborate model might include a third level where industry's production decisions were taken into account in more detail.

The objective (1.6) of the agricultural sector, which is the second level in the model, is to maximize its profits. It is assumed that as long as farm revenues exceed costs the set-aside land will be used for nonfood production. The problem for farmer f then is to decide how much of his land should be devoted to food crop c (x_{cf}), how much to nonfood crop d (xn_{cf}), and how much should remain fallow (xf_{cf}). It should be noted that a subsidy negotiated by the EU is paid to all farmers for each crop. For food crop c, this value is taken into account in the calculation of the gross margin, m_{cf}, appearing in the first term of (1.6); for nonfood crop d, this value, denoted by s_d, appears explicitly in the second term of (1.6). The third term represents income from direct payments received by farmers for leaving a portion of their land fallow.

Constraint (1.7) limits production by farm with the agricultural sector deciding the best use of the available land given the price p_d for nonfood crop d and gross margin m_{cf} for food crop c. Eq. (1.8) assures that $\theta \times 100\%$ of the arable land is either set aside or used to grow nonfood crops. Inequality (1.9) enforces an upper limit on sugar beet production (c' is the corresponding index). Because sugar beets have the highest price supports they are one of the most profitable crops and must be treated separately in the model. The final set of constraints (1.10) reflects agronomic considerations. The index k includes individual crops and groups of crops such as cereals. Each constraint limits output of the referenced crops on a particular farm f. For example, the output of oil crops, which comprise rapeseed, sunflower, and peas in the food category, and rapeseed and sunflower in the nonfood category, must not exceed 43.45% of the arable land. Note that the lower-level problem decomposes by farm. Although no attempt was made to exploit this fact, doing so might lead to an improvement in our algorithmic approaches.

3. Model extensions

An expansion of model (1) would include constraints that public sector decision makers might be required to take into account. For example, the 1992 Blair House Agreement (associated with the General Agreement on Tariffs and Trade – GATT) limits rapeseed and sunflower meal, a co-product of oil production that competes with Soya meal for cattle feed, to 1,000,000 tonnes annually. France's allocation is 600,000 tonnes. Modeling this constraint would require that the parameters $o_{\text{rapeseed, ester}}$ and $o_{\text{sunflower, ester}}$ assume one of two values, depending on whether the output of ester was above or below 600,000 tonnes. On the price side, the production of biofuels in substantial quantities may affect the market price of competing and complementary products. Nonlinear relationships worth exploring may also exist between conversion costs and the market price of co-products.

An expanded formulation for the industrial sector would allow for the possibility of producing all biofuels from all nonfood crops, something that is not technologically practical at present. To model this situation it is necessary to introduce a binary variable z_{db} equal to 1, if nonfood crop d is used to produce biofuel b, and 0 otherwise. Constraints (1.4) and (1.5) would now become

$$p_d \leqslant (\tau_b + v_b - \beta_{db} - \pi_b + o_{db})\alpha_{db} + \mu(1 - z_{db}) \quad \forall d \in D,\ b \in B(d), \tag{1.4a}$$

$$p_d \leqslant \mu \sum_{b \in B(d)} z_{db} \quad \forall d \in D, \tag{1.4b}$$

$$p_d \geqslant 0, \quad \tau_b \geqslant 0, \quad z_{db} \in \{0,1\}, \quad \forall d \in D,\ b \in B(d). \tag{1.5}$$

The new term on the right hand side of (1.4a), containing the setup variable z_{db}, is needed to deactivate the inequality when crop d is not used to make biofuel b. Constraint (1.4b) permits p_d to be positive (signaling to the agricultural sector a demand for crop d) only when industry agrees to buy crop d for production of at least one biofuel.

3.1. Alternative government–industry model

Different philosophies on the appropriate role that government should play in the economy give rise to different government objectives. Each may have widely different consequences for those who bear the costs and those who receive the benefits. With this in mind, we propose the following model that interchanges the roles of government and industry. It is assumed that a fixed budget in the form of tax credits, Θ, is available from the public treasury that industry can allocate as it likes subject to the previous constraints. The first portion of the new model is given below. The agricultural sector (1.6)–(1.11) remains the same.

Industry model (*leader*)

$$\max \sum_{b\in B}\sum_{f\in F}\sum_{d\in D(f)} \pi_b \alpha_{db} r_{df} xn_{df} \tag{2.1}$$

subject to

$$p_d \leqslant (\tau_b + v_b - \beta_{db} - \pi_b + o_{db})\alpha_{db} + \mu(1 - z_{db}) \quad \forall d \in D,\ b \in B(d), \tag{2.2}$$

$$p_d \leqslant \mu \sum_{b\in B(d)} z_{db} \quad \forall d \in D. \tag{2.3}$$

(*Government model*)

$$\sum_{b\in B}\sum_{f\in F}\sum_{d\in D(f)} \tau_b \alpha_{db} r_{df} xn_{df} \leqslant \Theta, \tag{2.4}$$

$$p_d \geqslant 0, \quad \tau_b \geqslant 0, \quad \pi_b \geqslant 0, \quad z_{db} \in \{0,1\}, \quad \forall d \in D,\ b \in B(d). \tag{2.5}$$

In model (2), industry's profit coefficient, π_b, is treated as a variable. Thus the leader's problem is to maximize profits for the petro-chemical industry subject to the same constraints, (2.2) and (2.3), on the price of nonfood crop d at the farm gate and a government constraint on the level of tax credits (2.4). Note that this new constraint is nonlinear in the both the leader's and follower's variables, and may cause added difficulty in finding a solution.

4. Description of algorithms

Model (1) is a nonlinear bilevel program (BLP) and not easy to solve even for small instances. The only complicating nonlinearity, though, appears in the leader's objective function as the cross-product term, $xn_{df} \times \tau_b$ (surface area allocated to nonfood crop d on farm $f \times$ tax credit for biofuel b). The cross-product term in the follower's objective function, $p_d \times xn_{df}$ (price at farm gate of nonfood crop $d \times$ surface area allocated to nonfood crop d on farm f) is of little consequence because once p is chosen by the leader, all that the follower must do is solve a linear program. The additional fact that the follower's constraint region is independent of the leader's decision variables simplifies the overall solution process.

The data set that we are working with contains 393 farms (Appendix A contains parameter data). Each farmer can grow up to 7 food crops and 5 nonfood crops. The available options are defined in sets C and D.

$$C = \{\text{wheat, barley, corn, sugar beet, rapeseed, sunflower, peas}\},$$

$$D = \{\text{wheat, corn, sugar beet, rapeseed, sunflower}\}.$$

At present, we are considering only two types of biofuels, ethanol and ester, that give rise to the following conversion sets:

$$D(\text{ethanol}) = \{\text{wheat, corn, sugar beet}\},$$

$$D(\text{ester}) = \{\text{rapeseed, sunflower}\}.$$

Today about a half-dozen computer codes exist for solving the linear bilevel programming problem (e.g., see Bard and Moore, 1990; Bialas and Karwan, 1984; Hansen et al., 1992; Júdice and Faustino, 1992). At best, they can handle 100 leader variables and 100 follower variables and 50 constraints. When nonlinearities are present, the manageable problem size shrinks by nearly an order of magnitude (Bard, 1988; Edmunds and Bard, 1991; Tolwinski, 1981). Our problem has 7 leader variables (level of tax credits for ethanol and ester; prices for the 5 nonfood crops), 3628 follower variables, 7 leader constraints and 3230 constraints in the agricultural sector model (1.6)–(1.11). This is much too big for any standard algorithm to solve. We have therefore taken an ad hoc approach and developed two distinct procedures that are shown to provide near-optimal solutions for the given application and accompanying data. This is established by extensive enumeration.

The basic idea in either case is to exploit the fact that once the biofuel tax credits are specified, the prices at the farm gate for nonfood crops can be readily computed from Eq. (1.4). Given these prices, the agricultural sector model which was formulated and coded in GAMS (Brooke et al., 1992), reduces to a linear program (LP). We note that GAMS is used to solve this LP at the first iteration of our solution algorithms only. OSL (1995) is used independently of GAMS to solve all subsequent subproblems.

4.1. Government model

The industry sector in model (1) is used to determine prices at the farm gate for the nonfood crops. The computations depend on industry's expected profit, conversion costs, and market prices for biofuels and co-products. For algorithmic purposes, we can rewrite (1.4) as

$$p_d = \max\left[(\tau_b + v_b - \beta_{db} - \pi_b + O_{db})\alpha_{db}\colon b \in B(d)\right] \quad \forall d \in D, \tag{1.4$'$}$$

where $D = \{$wheat, corn, sugar beet, rapeseed, sunflower$\}$; B(wheat) $= B$(corn) $= B$(sugar beet) $= \{$ethanol$\}$ and B(rapeseed) $= B$(sunflower) $= \{$ester$\}$ in the current data set. Thus there is a unique relationship between crop conversion and biofuels; that is, $|B(d)| = 1$ for all $d \in D$. This means that the 'max' operator in Eq. (1.4′) can be ignored. If a particular nonfood crop d could be converted into more than one biofuel b, then the 'max' operator would have to be used to compute the value of p_d.

4.2. Grid search algorithm (GSA)

In model (1), the leader (government) has control over τ_b and p_d; however, once τ is chosen p can be computed from Eq. (1.4′) so it is possible to view p as a function of τ; that is, $p = p(\tau)$. This relationship and the fact that $B = \{$ester, ethanol$\}$, implies that there are only two independent variables, a small enough number to impose a grid over their defined ranges and solve the government and farm models sequentially. Once we have found values for nonfood crop prices and the surface area allocated to each such crop, we can evaluate the government's two constraints (1.2) and (1.3) to determine whether or not the solution is feasible to the overall problem.

The range of the grid search for τ_{ester} and τ_{ethanol} is as follows. The basic steps of the algorithm are presented below.

$$0 \leqslant \tau_{\text{ester}} \leqslant 230 \text{ FF/hl},$$

$$0 \leqslant \tau_{\text{ethanol}} \leqslant 330 \text{ FF/hl}.$$

Implementation

1. We begin with a step size of 25 (denoted by STEP) and, for each point on the grid, compute the prices for corn, wheat, sugar beets, rapeseed, and sunflower using Eq. (1.4′). Next we solve the farm model (1.6)–(1.11) with these prices and compute the values for the government objective (1.1), surface constraint (1.2) denoted by *surf*, and ethanol production limit (1.3) denoted by *ethlim* (there is no practical limit on ester production). All points that are feasible with respect to the two constraints are stored in a table and sorted in ascending order by the value of the government objective (*gobj*).
2. A candidate list is then constructed by marking (i) the best point in the list (first), and (ii) the best point in each succeeding group of 10 points.

3. The step size is reduced from 25 to 5 and a new grid is constructed around each point on the candidate list. The grid is centered at the current value of τ_b, $b \in$ {ethanol, ester}, and is defined over the interval [τ_b – STEP_OLD, τ_b + STEP_OLD]. Points are enumerated sequentially and evaluated as in Step 1 above. Solutions that satisfy constraints (1.2) and (1.3) are added to the table of feasible grid points.
4. Steps 2 and 3 are repeated with STEP = 1. The algorithm then terminates.
5. The full table of feasible points is re-sorted by *gobj* and stored in an external file.
6. All points generated (feasible and infeasible) together with the government objective and constraint values, and attendant prices are written to an external file in a format suitable for spreadsheet importation.

In general, the effectiveness of the above procedure depends on the size of the grid and the number of variables over which the search has to be conducted. The finer the divisions, the more reliable the results. In our case, the fact that only two variables have to be fixed at every iteration allows us to explore the potential solution space in great detail. The speed at which the associated LPs can be re-optimized provides further advantage to the method. As the number of upper level variables grows, however, the computational burden grows geometrically and eventually undermines the reliability of the approach. In our implementation, the algorithm was terminated after the computations for a step size of 1 were completed. We felt that going any further would have yielded results incompatible with the quality of the input data. In real-world problems such as this, many coefficients such as conversion costs, market prices, and crop yields are at best estimates or averages, over time. Refining the calculations beyond the accuracy with which these coefficients are known will not produce better results.

4.3. Global convergence of GSA

A question naturally arises as to whether GSA will actually find a global solution to model (1) for a "sufficiently" small grid. Referring to model (3.1)–(3.4) below, the answer is suggested by Falk and Liu (1995) who show that for more general bilevel programs, the leader's objective function in the form $F \equiv F(\tau, p, x^*(\tau, p))$ is locally Lipschitz under certain assumptions; i.e., that the optimal solution, $x^*(\tau, p)$, to the follower's problem (3.3) and (3.4) for τ and p fixed satisfies strict complementary slackness (SCS), the strong second order sufficient condition (SSOSC) (i.e., the Hessian of the Lagrangian is strictly positive definite on the appropriate subspace), and linear independence (LI) of the binding constraints. These assumptions imply that $x^*(\tau, p)$ is locally Lipschitz and hence so is $F(\tau, p, x^*(\tau, p))$. Thus F is bounded in a neighborhood of (τ, p). In our case, the follower's problem is a linear program which may have alternative optima for some (τ, p). This implies that $x^*(\tau, p)$ is not necessarily Lipschitzian so neither is $F(\tau, p, x^*(\tau, p))$. It may be concluded, then, that GSA as currently implemented may not converge to a global optimum (Dempe, 1998).

Nevertheless, F is locally Lipschitz continuous at points where it is uniquely determined; i.e., on so-called regions of stability, which are the sets of all parameters (leader variables) where the optimal solution to the follower's problem does not change. In our case, we have only finitely many vertices of the feasible set of the follower's problem because this set is independent of the parameters (τ, p). Hence there are only a finite number of regions of stability; each takes the form of a polyhedron and is a convex, closed set. Restricted to one of these regions, the equivalent problem of minimizing $F(\tau, p, x^*(\tau, p))$ subject to $g(\tau, p, x^*(\tau, p)) = 0$ reduces to a linear program, at least implicitly (Lipschitz continuity in the objective function of the leader implies that it reduces to a Lipschitz continuous problem). Although we do not have the full function $x^*(\tau, p)$, we can compute a local approximation of it in the form of the Jacobian which can then be used to compute a direction of descent. Solving the bilevel program means that we have to solve a finite number of such linear programs (Dempe, 1998). This idea is pursued in the next section.

To guarantee global convergence of the modified version of GSA suggested above, the Lipschitz continuity on the region of stability of the lower-level optimal solution can be used to exclude parts of the regions of stability. More specifically, assume that we have an incumbent best point (τ^0, p^0) with corresponding objective function value F^0. Now consider a point (τ^1, p^1) with a lower-level solution x^1 and a Lipschitz constant $L1$ on the region of stability of the point x^1. Using these data, it is possible to exclude from further consideration that portion of the region of stability of x^1 in the set

$$\{(\tau, p) \text{ in the region of stability of } x^1\text{: } \|\tau - \tau^1\| + \|p - p^1\| \leqslant (F(\tau, p, x^1) - F^0)/L1\}.$$

However, we have to be careful: parameter values in other regions of stability can only be excluded with the help of optimal solutions associated with those values.

4.4. Nonlinear programming approach

Recall that in the formulation of the model, the government is given control over the decision variables $\tau \in T \subseteq \mathbb{R}^{n1}$ and $p \in P \subseteq \mathbb{R}^{n2}$, while the agricultural sector collectively controls the vector $x \in X \subseteq \mathbb{R}^{n3}$. The government goes first and attempts to minimize its objective function $F(\tau, p, x)$ over a feasible region defined by a set of functions in all problem variables. Because the government's objective function also depends on the farm sector's decisions, the former must anticipate each response or reaction of the follower before selecting a policy. Once the government makes a decision, the agricultural sector is faced with a traditional optimization problem of maximizing its objective function $f(\tau, p, x)$ over a feasible region $\{x \in X\text{: } h(\tau, p, x) \geqslant 0\}$ which is partially defined by τ and p. In our case, though, the constraint set of the follower is independent of the leader's policy variables so $h(\tau, p, x) = h(x)$.

In simple mathematical terms, problem (1) can be written as

$$\min_{\tau \in T,\ p \in P} F(\tau, p, x) \tag{3.1}$$

$$\text{subject to } g(\tau, p, x) \geqslant 0, \tag{3.2}$$

$$\max_{x \in X} f(\tau, p, x) \tag{3.3}$$

$$\text{subject to } h(x) \geqslant 0, \tag{3.4}$$

where T, P and X place additional restrictions such as bounds on the decision variables, $g\text{: } \mathbb{R}^{n_1+n_2+n_3} \rightarrow \mathbb{R}^{m_1}$, and $h\text{: } \mathbb{R}^{n_3} \rightarrow \mathbb{R}^{m_2}$. For the current data set, $m_1 = 7$ and $m_2 = 3628$.

In bilevel programming, it is customary to view the follower's problem as parameterized in the leader's variables. We can thus write $x = x(\tau, p)$ and rewrite the leader's objective and constraint functions as $F(\tau, p, x(\tau, p))$ and $g(\tau, p, x(\tau, p))$, respectively. Unfortunately, the vector $x(\tau, p)$ which is returned from the solution of the follower's problem (3.3) and (3.4), is not necessarily differentiable everywhere or even continuous in τ and p (see Shimizu et al. (1997) for a general discussion of nondifferentiability in bilevel programming). This makes it difficult to apply NLP theory to the BLP (3) directly, hence the development of the grid search algorithm. Nevertheless, some progress can be made if we are willing to sacrifice theoretical rigor in favor of an "engineering" approach.

In particular, we note that although the government's problem depends on the decisions of individual farmers to allocate surface area to specific crops, the real dependence is on crop prices, which we have seen are determined from the biofuel tax credits. The implication is that $p = p(\tau)$. Going one step further, the farm sector LP may be viewed as a function or subroutine that maps crop prices into allocation decisions. This dependence can be expressed as $x = x(p)$ without explicit refer to τ. The government model may then be formulated as a standard NLP with functions of the form $F(\tau, x(p))$ and $g(\tau, x(p))$; i.e.,

$$\min_{\tau,p} F(\tau, x(p)) \tag{4.1}$$

subject to

$$g_1(\tau, x(p)) \geqslant 0 \tag{4.2}$$

$$g_2(\tau, x(p)) \geqslant 0 \tag{4.3}$$

$$p_d - p_d(\tau_b) = 0, \quad \forall b \in B,\ d \in D(b) \tag{4.4}$$

$$p_d \geqslant 0, \quad 0 \leqslant \tau_b \leqslant \tau_{\max}, \quad \forall d \in D,\ b \in B, \qquad (4.5)$$

where (4.2) and (4.3) correspond to the government's constraints (1.2), (1.3) and (4.4) is equivalent to (1.4′). This formulation has the virtue of having only 7 decision variables (τ_b, $b \in$ {ethanol, ester}; p_d, $d \in D =$ {wheat, corn, sugar beet, rapeseed, sunflower}) and 7 constraints so it can be handled easily by any NLP solver.

4.4.1. General issues

All widely-used NLP solvers assume that all problem functions possess continuous first derivatives with respect to the decision variables (Lasdon et al., 1996). These derivatives are used to determine directions of movement and whether the optimality conditions are satisfied. As mentioned above, $F(\tau, x(p))$ and $g(\tau, x(p))$ are not continuously differentiable in τ and x. These variables depend on the crop prices in a discontinuous manner, varying as the farm sector LP changes bases. For example, when a particular p_d is zero or small, output of nonfood crop d will be zero. As p_d increases beyond some threshold, output will jump to a positive level on a subset of farms. This jump corresponds to a basis change and demonstrates the discontinuous nature of $x(p)$.

To apply an NLP code to model (4), the simplest approach is to ignore the discontinuities until they become too troublesome. NLP codes generally default to estimating first derivatives by finite differences, computing an average rate of change in each function over a small change in each variable. The best that can be hoped for in our case is to compute an average rate of change in x for a change in p that involves at least one basis change. Such estimates may or may not be strong enough to drive an NLP solver to a local optimum with any reliability. Here 'reliability' is taken to mean that for different instances of the same model (different data) a point acceptably close to a local optimum can be found. Indication that the derivative estimates are well-behaved would be that the solver makes reasonably steady progress and routinely terminates when the fractional change in the objective value is below some predetermined threshold. When progress has stalled or the line search algorithm repeatedly fails, the natural conclusion would be that the derivative information is no longer useful.

To solve (4) with an NLP code, gradient information is needed with respect to the decision variables τ and p. Two options are available for obtaining this information from the farm sector LP:

1. Determine a single finite difference perturbation factor for the prices that will induce at least one basis change. This is relatively simple to implement since many NLP codes allow the user to set the perturbation factor; in others the value is easily set in the source code.
2. A more precise method would be to exploit the sensitivity information in the LP solution by determining for each price, the minimum change required to induce a basis change. This approach would require that the differentiation routine in most NLP solvers be modified to allow a different perturbation step for each variable.

If neither of these methods results in reliable performance on the part of the NLP solver, it is reasonable to conclude that the inherent discontinuities in the problem cannot be ignored. In our implementation we took the first approach and experimented with the standard codes GRG2 and SQP (Fan et al., 1988) as the NLP solver. The latter turned out to be more suitable for our problem.

4.5. Formulating the follower's problem as a QP

In light of the uncertainty and limitations associated with obtaining derivative estimates for SQP with the LP follower problem, the question arises as to whether a more reliable and robust procedure might be developed. Addressing this issue, we note that the gradient of F with respect to (τ, p) is given by

$$\nabla F(\tau, x(p)) = [\nabla_\tau F(\tau, x(p)),\ 0] + [0,\ \nabla_x F(\tau, x(p)) \nabla_p x(p)],$$

where $\nabla_p x(p)$ is the Jacobian of $x(p)$, denoted by $J(p)$, and the ∇ operator produces a row vector. As mentioned above, $x(p)$ is not, in general, differentiable with respect to p, or even known, so finding $J(p)$ is problematic (Kiwiel, 1995). To circumvent this difficulty we can add the smoothing term $-\epsilon\|x - x_0\|^2$ to the follower's objective function, where x_0 is a fixed reference point and $\epsilon > 0$ is a small constant. This leads to a *regularized* version of the follower's problem in which we would have to find

$$x(p) = \operatorname{argmax}\Big(f(x,\ p) - \epsilon\|x - x_0\|^2 : h(x) \geqslant 0,\ x \in X\Big)$$

and an element of $J(p)$. An "easy" case arises when strict complementarity holds and the active constraints are linearly independent, i.e., $x(\cdot)$ is differentiable at p. Then $J(p)$ can be computed via the implicit function theorem from the Kuhn–Tucker conditions for the associated equality constrained quadratic program (EQP) subproblem in which the inactive constraints are ignored and the active ones are treated as equalities. The remaining "hard" case is messy but again an engineering approach can be used. Specifically, the EQP subproblem may be derived by ignoring constraints with null Lagrange multipliers and those linearly dependent on the remaining ones. Then $J(p)$ may be computed from EQP as before.

We attempted to implement in some measure the ideas described above by augmenting the existing LP objective with a quadratic penalty term giving

$$\max\left(c^{\mathrm{T}}x - \epsilon\|x - x_0\|^2\right), \tag{5}$$

where x_0 is the previous point at which the QP was solved. We were able to do this by exploiting OSL's facility for *layering* a QP on top of an existing LP model. Specifically, that facility allows the user to solve QPs of the form

$$\max\ c^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Qx$$

$$\text{subject to } Ax \leqslant b$$

by specifying the quadratic matrix Q then invoking the QP solution algorithm. We specified Q to be $-2\epsilon I$, computed the coefficients for the linear terms, and then added them to the corresponding coefficients in the LP objective.

As an aside, we note that Falk and Liu (1995) provide a related solution approach. They first propose a method to compute directional derivatives for $x(\tau,\ p)$ with respect to τ and p from the follower's problem. This information is used to compute subgradients of $F(\tau,\ p,\ x(p))$ with respect to τ and p. A bundle algorithm is then devised to solve the leaders problem. They show that the overall methodology converges to a regular point of the general BLPP under appropriate conditions.

4.6. Algorithm implementation

We denote by BIOFUEL the system that sets up and solves the agricultural model for a given set of prices and subsidies and evaluates the government objective and constraints. BIOFUEL loads the LP agricultural model from an MPS file generated by GAMS. It loads the remaining required problem structure information from a set of external files which are generated by programs that determine the GAMS problem structure from the GAMS listing file. For a full description of the algorithmic design, data structures and subroutines, see Bard (1999, Ch. 12).

Fig. 1 highlights the principal components of BIOFUEL. The top box, identified as the *central calling system*, is a reference to either the GSA or the SQP-based code. Both call GOVT_MODEL to compute prices and then the FARM_MODEL to solve to the corresponding mathematical program. OSL sits at the bottom of the system and repeatedly solves these subproblems.

To assist in the generation of derivative estimates, we allow the NLP solver to manipulate the prices as decision variables so that $x(p)$ does not depend implicitly on τ. Because the prices are in fact functions of the tax credits, we obtain from the government model subroutine, the computed values of the prices for a given set of τ values. Constraint (4.4) requires that the difference between the specified and computed prices be zero in any feasible solution.

In the design of BIOFUEL, we made provisions for any NLP code to be called to solve model (4).

The top level interface allows subsidies and prices to be passed down and objective and constraint values to be passed out. All setup relevant to BIOFUEL itself is performed internally at the first function evaluation call.

One important criterion for the choice of SQP was that it requires fewer function evaluations than many competing NLP algorithms and is hence attractive when those function evaluations are expensive, as is obviously the case here (the

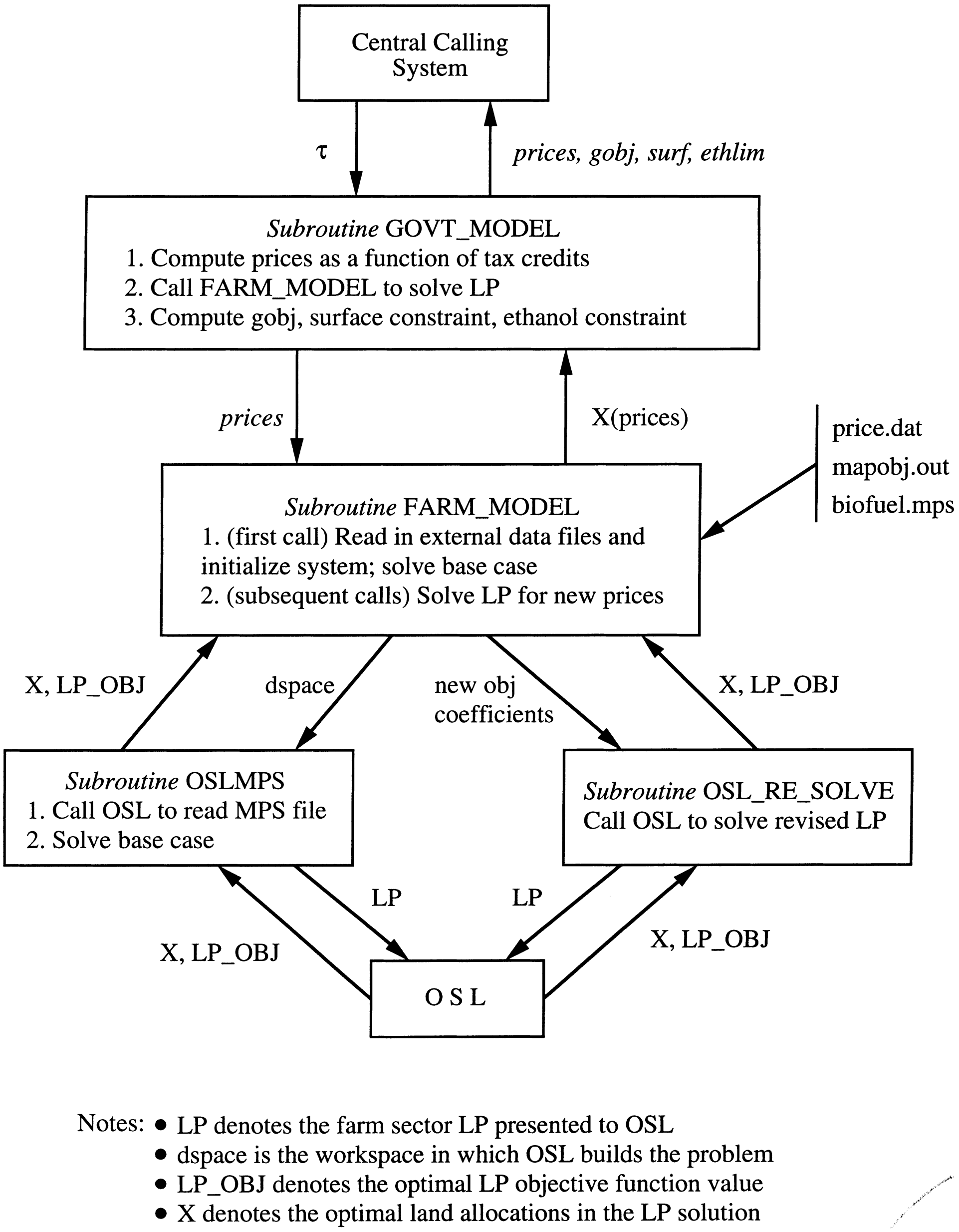

Fig. 1. BIOFUEL module structure and data flow.

agricultural sector model for the case examined has slightly in excess of 3600 variables).

All NLP solvers possess some set of parameters that can be used to *tune* the algorithm to the specific nature of the problem under consideration. We experimented briefly with arbitrary finite difference perturbation steps and attempted to fine tune the algorithm's parameters which control the deviation variable penalty weights, the fractional change stopping criterion, and the Kuhn–Tucker optimality tolerance. The optimality tolerance was set to zero, since the nondifferentiable nature of the problem makes this test meaningless. The algorithm should run until further progress is impossible.

5. Computational results

All algorithms have been coded in FORTRAN 77. The grid search and initial SQP runs with the LP follower problem were made on a SUN Sparcstation 10. The SQP runs with the LP follower problem (Table 2) and the corresponding runs with the QP follower problem (Table 3) were made on a Pentium 100 system using the WATCOM FORTRAN 77 V10.6 compiler.

5.1. Grid search output

GSA takes approximately 30 min to run on the SUN. A high level diagram of the basic routines is given in Fig. 2 where the call to the GOVT_MODEL refers to Fig. 1. At each iteration, the LP farm model with 3628 variables and 3230 constraints is re-optimized with OSL. The specifics are discussed in Bard (1998). The best solution obtained for a grid size of 25 occurs at

$$\tau_{\text{ester}} = 125, p_{\text{rapeseed}} = 400.5, p_{\text{sunflower}} = 347.8,$$
$$\tau_{\text{ethanol}} = p_{\text{wheat}} = p_{\text{corn}} = p_{\text{sugar\_beet}} = 0$$

with

$$gobj = 2.11 \times 10^7, surf = 4786.0 \text{ and}$$
$$ethlim = 3 \times 10^6.$$

The last value implies that ethanol is not produced.

After a series of refinements to the grid, first cutting the step size to 5 and then to 1 as indicated in Steps 3 and 4 of the algorithm, we get the best solution at

$$\tau_{\text{ester}} = 117, p_{\text{rapeseed}} = 365, p_{\text{sunflower}} = 310,$$
$$\tau_{\text{ethanol}} = p_{\text{wheat}} = p_{\text{corn}} = p_{\text{sugar\_beet}} = 0$$

with

$$gobj = 2.06 \times 10^7, surf = 48.8 \text{ and}$$
$$ethlim = 3 \times 10^6.$$

(In fact, the output indicates that $\tau_{\text{ethanol}} = 5$, but since there is no ethanol production we can interpret this as zero.) Table 1 presents a sampling of output in the neighborhood of this solution. The rows are sorted by the government objective value. As can be seen, the first two entries are not feasible because not enough surface area is being

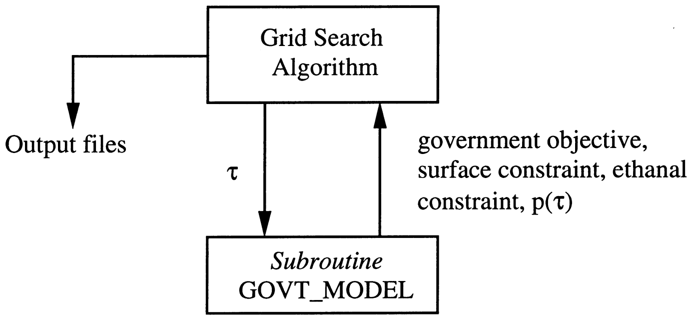

Fig. 2. Central calling structure for Grid Search Algorithm.

Table 1
Sample output from grid search algorithm

$\tau_{ethanol}$ (FF/hl)	τ_{ester} (FF/hl)	Govt. objective (10^5)	Farm objective (10^6)	Surface (ha)	Ethlim (10^6)
0	115	188.49	1918.688	−770	3.00
5	116	193.82	1918.854	−560	3.00
5	117	206.32	1919.026	49	3.00
260	115	209.96	1918.709	−499	2.99
5	118	224.75	1919.205	971	3.00
5	119	241.99	1919.403	1857	3.00
265	115	242.28	1918.768	−131	2.98
0	120	250.36	1919.608	2212	3.00
5	121	260.89	1919.822	2639	3.00
260	120	271.04	1919.628	2435	2.99
5	122	273.59	1920.041	3203	3.00
5	123	283.69	1920.267	3593	3.00
270	115	307.97	1918.931	635	2.96

farmed. In addition, there is no ethanol production until the tax credit reach about 260 FF/hl. A final point to note about the results is that the base case, with

$$\tau_{ester} = 230, \tau_{ethanol} = 330, p_{rapeseed} = 873,$$
$$p_{sunflower} = 841, p_{wheat} = 511, p_{corn} = 581, \text{ and}$$
$$p_{sugar\_beet} = 140,$$

yields

$$gobj = 1.05 \times 10^9, surf = 4.96 \times 10^5 \text{ and}$$
$$ethlim = -2.17 \times 10^5$$

which is not feasible. That is, ethanol is overproduced by a significant amount. Also, the corresponding cost to the government in about 50 times higher than the best solution found.

5.2. SQP output

A high level diagram of the SQP-based code is given in Fig. 3 where again the call to the GOVT_MODEL refers to Fig. 1. We examined the performance of this system from three starting points for τ_{ester}: (i) $\tau_{ester} = 10.0$ (far below the grid search optimum of 117); (ii) $\tau_{ester} = 125.0$ (in the neighborhood of 117); and (iii) $\tau_{ester} = 230.0$ (far above 117). Each employed the same initial prices ($p_{wheat} = 5.0$, $p_{corn} = 5.0$, $p_{sugar\_beet} = 5.0$, $p_{rapeseed} = 400.5$, $p_{sunflower} = 347.8$). The first three prices are a considerable distance from the optimum; the latter two are much closer. In any real-world instance, the current and past prices would be known, so any systematic examination of algorithmic performance for prices far away from the optimum or some known base point is difficult to justify. The second variable, $\tau_{ethanol}$, was started at 250 in each case.

Because of the large differences in magnitudes of the functions involved, we scaled the government objective by 10^{-9} and the surface and ethanol constraints by 10^{-4}. After some experimentation, we settled on an SQP penalty weight of 10.0 and a derivative perturbation step of 0.01. The results are encouraging though not definitive. The SQP runs generate a solution path that moves monotonically towards feasibility, obtains feasibility, then monotonically decreases the objective until several successive line searches fail and the algorithm terminates on the criterion ‘all remedies have failed to find a better point’. This behavior is about as much as can be expected. The process eventually runs out of usable derivative information and is unable to compute a direction of descent. Table 2 gives the results from each starting point for the SQP runs with the LP follower problem. In each case, the value of $\tau_{ethanol}$ converged to around 190 which is well below the trigger point of 260 for ethanol production. This implies that $\tau_{ethanol}$ can be taken as zero.

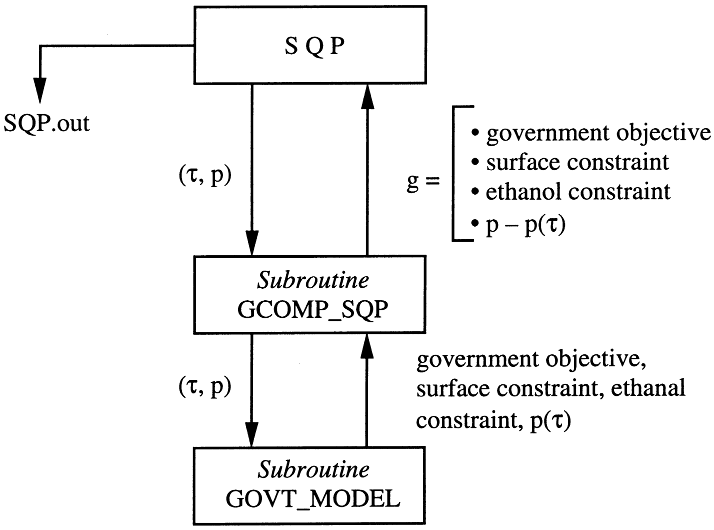

Fig. 3. Central calling structure for SQP.

We also ran SQP using the QP follower problem formulation over the same cases for three different values of the penalty weight $\epsilon \in \{10^{-10}, 10^{-6}, 10^{-4}\}$ in (5). For the first two values of ϵ, the results were identical to those in Table 2. The results for $\epsilon = 10^{-4}$ are given in Table 3. The corresponding CPU times are somewhat exaggerated as we had difficulties with the warm start option for OSL's QP solver. For reasons that we have yet been able to characterize, the QP warm starts periodically locked up the Pentium, so all QP runs were cold started (including those solved during finite difference derivative computations). Fixing this problem should result in at least an order of magnitude improvement in CPU time, although the approach would still be slower by a factor of 2–4.

When started from $\tau_{\text{ester}} = 230$, both formulations yielded values almost identical to the best values obtained by the grid search; a similar statement can be made for the first formulation for $\tau_{\text{ester}} = 125$. In either case, the runs started from $\tau_{\text{ester}} = 10.0$ terminated at a slightly inferior point. Notably, both formulations yielded identical results for the first and third starting points. If the QP formulation is generating better derivative in-

Table 2
SQP runs with LP follower problem formulation

τ_{ester} (initial)	τ_{ester} (final)	Govt. objective	SQP iterations	CPU time (s)
10.0	117.67	2.08078×10^8	26	240
125.0	117.06	2.06415×10^8	20	191
230.0	117.05	2.06408×10^8	22	209

Table 3
SQP runs with QP follower problem formulation

τ_{ester} (initial)	τ_{ester} (final)	Govt. objective	SQP iterations	CPU time (s)
10.0	117.67	2.08078×10^8	26	10820
125.0	117.65	2.08060×10^8	17	7162
230.0	117.05	2.06408×10^8	22	9381

formation, it cannot be inferred from these runs since the LP formulation attains what appears to be the optimum with no greater algorithmic difficulties (e.g., line search failures) than the QP formulation. Examination of the detailed output confirmed that SQP followed the same path in either case.

6. Discussion

The main advantage of the grid search procedure is that it avoids the difficulties resulting from the nondifferentiability of the relationship between the LP decision variables (xn_{df}) and the nonfood crop prices (p_d). Any substantial increase in problem scale with respect to the number of biofuels or the number of nonfood crops would likely induce a combinatorial explosion. This would drastically reduce the efficiency and, ultimately, the viability of the procedure. For the problem instance we were faced with, however, characterized by a small number of upper level variables and constraints and a large number of lower level variables and constraints, this approach proved reliable and effective.

With regard to SQP, we note that it is not a feasible path algorithm so it may move into and out of the feasible region. For a problem such as this where the Kuhn–Tucker optimality test is not relevant and the solution process is expected to terminate by satisfying the fractional objective change criterion or with the 'all remedies failed' condition, it is quite possible that the final point will not be feasible. Therefore, in any real-world implementation it would be essential to store and update the best feasible point visited by the solver and ultimately report the incumbent as the solution.

A final and somewhat unsettling point about the NLP approach is that it has a number of weaknesses that cannot be brushed aside. Although we were able to find what appeared to be the optimal solution to our problem, this may not always be the case. It is quite possible that for different data or a different LP subproblem the algorithm will get bogged down immediately with bad derivative information and not be able to proceed. Should this occur, starting the NLP solver at different point might provide a fix. More testing would lead to a greater understanding of this issue.

Appendix A. Data set for subsidy model

Definition of data elements:

$\alpha(d,b)$	factor for converting one tonne of nonfood crop d to biofuel b (hl/t)
$\beta(d,b)$	cost of converting nonfood crop d to biofuel b (FF/hl)
$o(d,b)$	market price of co-products associated with production of one unit of biofuel b from nonfood crop d (FF/hl)
$\pi(b)$	profit expected by industry for one unit of biofuel b (FF/hl)
$v(b)$	market price for one unit of biofuel b (FF/hl)
$\tau(b)$	government tax credit given to industry for biofuel b (FF/hl)
D	{wheat, corn, sugar beet, rapeseed, sunflower} = set of nonfood crops
B	{ester, ethanol} = set of biofuels

Data for model ('no' indicates the conversion is not possible):

α(wheat,ester)	no
α(corn, ester)	no
α(sugar beet, ester)	no
α(rapeseed, ester)	4.5 hl/t
α(sunflower, ester)	4.7 hl/t
α(wheat, ethanol)	3.5 hl/t
α(corn, ethanol)	3.8 hl/t
α(sugar beet, ethanol)	1 hl/t
α(rapeseed, ethanol)	no
α(sunflower, ethanol)	no
β(wheat, ester)	no
β(corn, ester)	no
β(sugar beet, ester)	no
β(rapeseed, ester)	168 FF/hl
β(sunflower, ester)	168 FF/hl

β(wheat, ethanol)	207 FF/hl
β(corn, ethanol)	207 FF/hl
β(sugar beet, ethanol)	130 FF/hl
β(rapeseed, ethanol)	no
β(sunflower, ethanol)	no
o(wheat, ester)	no
o(corn, ester)	no
o(sugar beet, ester)	no
o(rapeseed, ester)	120 FF/hl
o(sunflower, ester)	105 FF/hl
o(wheat, ethanol)	83 FF/hl
o(corn, ethanol)	90 FF/hl
o(sugar beet, ethanol)	0
o(rapeseed, ethanol)	no
o(sunflower, ethanol)	no
π(ester)	60 FF/hl
π(ethanol)	120 FF/hl
v(ester)	72 FF/hl
v(ethanol)	60 FF/hl

Range on tax credits:

$0 \leqslant \tau(\text{ester}) \leqslant 230$ FF/hl
$0 \leqslant \tau(\text{ethanol}) \leqslant 330$ FF/hl

References

Aiyoshi, E., Shimizu, K., 1981. Hierarchical decentralized systems and its new solution by a barrier method. IEEE Transactions on Systems, Man, and Cybernetics, SMC-11 (6), 444–449.

Anandalingam, G., Friesz, T.L., (Eds.), 1992. Hierarchical optimization. Annals of Operations Research 34 (1–4).

Bard, J.F., 1988. Convex two-level optimization. Mathematical Programming 40, 15–27.

Bard, J.F., 1999. Practical Bilevel Programming: Algorithms and Applications. Kluwer Academic Publishers, Boston.

Bard, J.F., Falk, J.E., 1982. A separable programming approach to the linear complementarity problem. Computers & Operations Research 9 (2), 153–159.

Bard, J.F., Moore, J.T., 1990. A branch and bound algorithm for the bilevel programming problem. SIAM Journal of Scientific and Statistical Computing 11 (2), 281–292.

Bialas, W.F., Karwan, M.H., 1984. Two-level linear programming. Management Science 30 (8), 1004–1020.

Brooke, A., Kendrick, D., Meerhaus, A., 1992. GAMS: A User's Guide, Boyd and Frazer, Danvers, MA.

Dempe, S., 1998. Private communications. Faculty of Mathematics and Informatics, Freiburg University of Mining and Technology, Freiburg, Germany.

Edmunds, T.A., Bard, J.F., 1991. Algorithms for nonlinear mathematical bilevel programs. IEEE Transactions Systems, Man, and Cybernetics 21 (1), 83–89.

Falk, J.E., Liu, J., 1995. On bilevel programming, part I: General nonlinear case. Mathematical Programming 70, 47–72.

Fan, Y., Sarkar, S., Lasdon, L., 1988. Experiments with successive quadratic programming algorithms. Journal of Optimization Theory and Applications 56 (3), 359–383.

Fortuny-Amat, J., McCarl, B., 1981. A representation and economic interpretation of a two-level programming problem. Journal of the Operational Research Society 32, 783–792.

Hansen, P., Jaumard, B., Savard, G., 1992. New branch-and-bound rules for linear bilevel programming. SIAM Journal of Scientific and Statistical Computing 13 (5), 1194–1217.

Júdice, J.J., Faustino, A.M., 1992. A sequential LCP method for bilevel linear programming. In: Anandalingam, G., Friesz, T.L. (Eds.). Hierarchical Optimization. Annals of Operations Research 34 (1–4), 89–106.

Kiwiel, K.C., 1995. Private communications, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.

Lasdon, L., Plummer, J., Warren, A., 1996. Nonlinear programming. In: Avriel, M., Golany, B. (Eds.), Mathematical Programming for Industrial Engineers, Ch. 6, Marcel Dekker, New York, pp. 385–485.

OSL 1995, Optimization Subroutine Library: Guide and Reference, IBM, Corp., Dept. 55JA, Poughkeepsie, NY.

Shimizu, K., Ishizuka, Y., Bard, J.F., 1997. Nondifferentiable and Two-Level Mathematical Programming. Kluwer Academic Publishers, Boston.

Tolwinski, B., 1981. Closed-loop Stackelberg solution to multistage linear-quadratic game. Journal of Optimization Theory and Applications 34 (4), 485–501.