Parametric Optimization of Linear and Non-Linear Models via Parallel Computing to Enhance Web-Spatial DSS Interactivity

Article *in* International Journal of Decision Support System Technology · January 2012

DOI: 10.4018/jdsst.2012010102

CITATION
1

READS
175

3 authors:

Dimitrios Kremmydas
European Commission
34 PUBLICATIONS **142** CITATIONS
SEE PROFILE

Athanasios Petsakos
22 PUBLICATIONS **298** CITATIONS
SEE PROFILE

Stelios Rozakis
Technical University of Crete
120 PUBLICATIONS **1,112** CITATIONS
SEE PROFILE

Parametric Optimization of Linear and Non-Linear Models via Parallel Computing to Enhance Web-Spatial DSS Interactivity

D. Kremmydas, Agricultural University of Athens, Greece

A. Petsakos, Agricultural University of Athens, Greece

S. Rozakis, Agricultural University of Athens, Greece

ABSTRACT

A web based Spatial Decision Support System (web SDSS) has been implemented in Thessaly, the most significant arable cropping region in Greece, to evaluate selected energy crop supply. The web SDSS uses an optimization module to support the decision process launching mathematical programming (MP) profit maximizing farm models. Energy to biomass raw material cost is provided in supply curve form incorporating physical land suitability for crops, farm structure, and Common Agricultural Policy (CAP) scenarios. To generate biomass supply curves, the optimization problem is parametrically solved for a number of steps within a price range determined by the user. The more advanced technique used to solve the MP model, the higher the delay of response to the user. In this paper, the authors examine how effectively the web SDSS response time can be reduced to the user requests using parallel solving of the corresponding optimization problem. The results are encouraging, as the total solution time drops significantly as the problem's size increases, improving the users' experience even when the underlying optimization models use advanced, time demanding modeling techniques. These statements are illustrated by comparing lp and non-lp agricultural sector models.

Keywords: *Common Agricultural Policy (CAP), Farm Model, Parallel Programming, Positive Mathematical Programming (MP), Spatial Decision Support Systems (SDSS)*

INTRODUCTION

The progress in Web-based decision support technologies has been recently described by Bhargava et al. (2007) who distinguish between model-driven and data-driven decision support system (DSS) to provide an impressive list of systems for decision support using the web as a medium (stand-alone commercial applications) or as a computer (web-DSS). Most applications

DOI: 10.4018/jdsst.2012010102

concern business decision support, whereas some deal with environmental issues involving also multi-criteria models often attempting to enhance public participation in local environmental decision making (Kingston et al., 2000). One of the most interesting classes of web-based decision support tools are the so-called Spatial DSS (SDSS). SDSS, as defined by Sugumaran and Sugumaran (2007), are "flexibly integrated systems built on a GIS platform to deal with spatial data and manipulations, along with an analysis module ... they support 'what if' analysis ... and help the user in understanding the results" (p. 850). With the development of the internet, Web-based SDSS have been developed, adding Internet interface programs to the computational models and geographic databases of the SDSS, in order to provide decision support through the Web based on relevant information.

Bio-energy issues constitute by excellence spatially dependent problems requiring both detailed spatial information but also extensive model building. Unlike conventional energy carriers that have hierarchical structure, biomass-to-energy production involves hundreds to often thousands of decentralised decision makers. This is considered one of the "grand challenges" for bio-energy assessment (McKone et al., 2011). As a matter of fact, bio-energy profitability is linked to the structure and perspectives of the arable cropping systems to supply considerable quantities of a bulky raw material to transformation plants also taking into account demand location and volume. Recent analyses of economic biomass potential are reported in regional (Hilst et al., 2010) or country level (Simon et al., 2010). Therefore, appropriate tools are necessary to enable comprehensive analysis and support decisions of policy makers, industry, researchers and farmers. For this purpose, a state-of-the-art modular SDSS that contains optimization models embedded in a GIS environment fed by technical, economic, and cartographic databases has been built to provide stakeholders with region specific biomass-to-energy supply information in Central Greece (Rozakis, 2010). A web-based interface built in open source software makes the SDSS tool available for collaborative decision-making allowing for an interactive process in real time. The tool operates on the Internet, where the user can have access to the dataset, enter selected parameters into the model, and enable spatial visualization and exploration of the results, injecting interactivity in the decision process.

In order to adequately represent arable agriculture of the region under study, bottom-up mathematical programming models have been used to estimate agricultural policy impacts and farmers' supply response. Numerous gross margin maximizing Decision-Making Units (DMU), geographically dispersed decide whether or not to introduce energy crops in their crop mix using crop suitability maps and survey data at the farm level. Conventional linear programming (LP), traditionally used for this purpose, is gradually being dominated in the agricultural economics literature by alternative methods implemented also in the Greek context as multi-criteria (Manos et al., 2009) or interval linear programming (Rozakis, 2011) models and also positive models incorporating downward sloping demand (Rozakis et al., 2008) or increasing cost functions (Petsakos & Rozakis, 2010) in the objective function. These methods, broadening economic rationality, manage to transform the objective function so that optimal solutions include not only crop plans on the vertices of the feasible polyhedron but also points on hyper-plans enabling the model to approach observed levels of activities, thus outperforming their LP counterparts. Nevertheless there is a price to pay that is the increased complexity and consequently solution time span of such models. That may not be a problem when models are operated for research purposes, but it certainly is a serious drawback in business or policy-making oriented environments and especially in a context of interactive decision making such as the one previously described.

Regional farm based sector models articulated in an angular structure are parametrically solved to explicit supply response to bio-energy market signals, in other words optimization is consecutively launched for different entry data

(data related models). Approximately resolution times are multiplied by the number of iterations. This penalizes nonlinear models since they require longer solving times than the corresponding linear model. However the modular structure of bottom-up arable agriculture models consists of numerous independent problems that constitute an embarrassingly parallelizable problem as each iterative solution is independent to anyone else. This feature makes the parallel solving of such problems quite interesting since lapse time for resolution is drastically reduced. Furthermore, the extensive use of Personal Computers (PCs) within the scientific community and tremendous increase in their CPU's frequency, and the advent of multi-core CPUs and network technologies (intranets and internet) has rendered distributed computing infrastructures readily accessible even to modest research institutes (Creel, 2005).

Parallel computing is implemented in this paper aiming at improving efficiency of the optimization process in the bio-energy assessment web-SDSS. Three model types are used namely an LP model and two nonlinear specifications that represent different degrees of structural complexity. The next section introduces the concept of parallel computing in the case of web accessible Decision Support Systems. The methodology of the optimization component and the model specification for arable agriculture in Thessaly, Greece, is then presented. Model parallelization, the implementation issues and the speedup results for a case study of integrating a web-SDSS with a parallel LP meta-solver follow. The paper is completed by concluding remarks and issues of further research work.

PARALLEL COMPUTING FOR DATA RELATED MODEL-DRIVEN WEB-DSS

Web-Based DSS deliver decision support information or decision support tools using a "thin-client", that is a Web browser. A model-driven web-DSS such as the one supporting biomass assessment, according to the typology of Power (2004) "uses formal representations of decision models and provide analytical support using tools of decision analysis, optimization, stochastic modeling, simulation, statistics and logic modeling". A model-driven web-DSS should contain at least two components: The user interface component, which would be some kind of web application and the decision analysis component that would include the necessary software that will perform the decision analysis. The former component is the front-end which the user interacts with the web-DSS by feeding input to the latter component and obtaining results from it.

Tolerable waiting time (TWT) is defined as the amount of time users are willing to wait before giving up on the download of the web page. There are several papers that attempt to measure TWT with time spans ranging from 4 to 41 seconds (Nah, 2004). For a web-DSS the above time values should not be considered literally, since the user is more dedicated to the purpose of obtaining the results (that is downloading the web page) than a user browsing or querying various sites. However the above range gives us an order of magnitude of the time a web-DSS system should respond and that it should not exceed one minute. Also it is deducted that for the same web-DSS, as the waiting time decreases, the user experience is improved and enriched.

Given the high possibility that the computation procedures might be a major source of delaying the system's response, we are looking for ways to decrease this delay. Implementing parallel computing solutions to our decision analysis can decrease the total waiting time for the user, thus moving the overall performance of the system towards a tolerable waiting time.

There are cases where solving the decision problem in parallel is embarrassingly easy, for example when the decision process incorporates solving a Monte Carlo simulation, performing sensitivity analysis, solving different scenarios or when we have to solve multiple independent linear problems (data related problems). The models that we are using on our web-SDSS

(regional sector models in a farm basis) fall to the latter category. Migrating from an existing (serial) decision analysis component of a web-DSS to a parallel solution is not a trivial task since several issues have to be resolved. For example we are primarily concerned about the immediate distributed resources availability. Utilizing a batch processing distributed system is very likely to bring on delays. There is also an issue about the cost-benefit ratio of migrating to a parallel solution. The costs of adapting the serial implementation of the decision analysis process to a parallel system can be significant and for example it could include the development and the deployment of the software solution, the maintenance costs of the cluster, etc. On the other hand the benefit of using a parallel system is the decrease in the user waiting time, and this is greater as the problem size is increasing.

METHODOLOGY FOR ESTIMATING BIOMASS-TO-ENERGY SUPPLY

Traditional Programming Techniques for the Agricultural Sector

In the agricultural economics literature as Kutcher and Norton (1982) narrate, analysts tackled with the farm decision problem that is maximizing profits under constraints, aiming either to deriving the best plan or more usually to assist policy decision by illuminating the varied consequences of multiple choices. This kind of analysis was carried out by means of mathematical programming. At first such models simulated producers responses searching the optimal plan assuming parameters such as output prices fixed, specified in linear programming form.

The elementary sub-model, is specified as follows: an individual farmer (f) is supposed to choose a cropping plan $(\mathbf{x}^f)$ and input use among technically feasible activity plans $\mathbf{A}^f\mathbf{x}^f \leq \mathbf{b}^f$ so as to maximize gross margin, gm^f. The optimization problem for the farmer f appears as seen in Box 1.

The sector model contains f farm problems such as the one specified above. The basic farm problem is linear with respect to $\mathbf{x}^f$, the primal $n \times 1$ vector of the n cropping activities. The $m \times n$ matrix $\mathbf{A}^f$ and the $m \times 1$ vector $\mathbf{b}^f$ represent respectively the technical coefficients and the capacities of the m constraints on production. The vector of parameters $\boldsymbol{\theta}^f$ characterizes the *f*-th representative farm and includes yields for crop i (y_i^f), variable costs (c_i^f), prices dependent on quality (p_i^f) and subsidies linked to crop quantity (ls_i^f). Symbol $\boldsymbol{\kappa}$ stands for the vector of general economic parameters which includes prices not dependent on farm (p_i) and subsidies specific to crop cultivated area (ps_i). The constraints can be distinguished in resource, agronomic, demand and policy ones. The model enables a comparative static analysis, but does not allow for farm expansion, as it takes as given land resource endowments and land rent of the base year. Different sets of parameters are applied to denote the policy context in vigor.

Unlike the standard LP formulation, where input and output prices are assumed fixed and exogenous, price endogenous models are used in situations where this assumption is flawed or untenable. Consequently, LP models can be modified to include a response of the market (i.e., of the aggregate of producers) under the assumptions regarding market form and producer decision rules leading in non-linear programming (NLP) specifications (Bauer & Kasnakoglou, 1990). For instance, it is usual that the quantity of fodder crops produced affects the equilibrium price primarily due to the high transportation costs which restricts its consumption locally or to adjacent regions. As a result, and given the limited alternative uses of fodder crops, the analyst assumes that the price received by producers is determined by the total amount produced in the region. Price endogenous module for fodder crops renders the model quadratic belonging to the class of NLP.

Table 1. Aggregates of different model specifications and observed rotation for 2002 (ha)

	Observed	LP	NLP	PMP
Soft Wheat	396.5	2935.9	2935.9	396.5
Durum Wheat	52039.7	52781.8	52599.8	52039.7
Maize Grain	14906.9	20620.6	18661.4	14906.9
Maize Fodder	366.9	0.0	0.0	366.9
Tobacco	1833.6	1833.6	1833.6	1833.6
Cotton	166886.5	164871.7	165476.5	166886.5
Potato	123.8	70.5	70.5	123.8
Sugar Beet	12806.0	9830.8	9262.5	12806.0
Tomato	556.0	412.64	412.6	556.0
Alfalfa	5899.7	2458.0	4562.7	5899.7
AAD		1799.1	1370.8	0.0

Model Validation

A validation process is always required in order to assess model's ability to predict farmers' response to different market signals or policy shifts. For this purpose, observations for base year are compared to model results by examining appropriate distance measures. Among them the average absolute deviation (AAD) index is readily used, defined as the average absolute difference between the observed data and the land allocations generated by the model at the optimum: $\text{AAD} = \frac{1}{I}\sum_{i=1}^{I}\left|x_i^{model} - x_i^{obs}\right|$, where i is the crop index (with $i = 1,2,\ldots,I$).

In order to use a model for any kind of prediction such as to generate reliable biomass supply curve to be exploited by the industry, different model specifications are validated so that the most efficient to approach the initial situation to be selected. Among the above mentioned specifications, NLP usually results in much lower AAD index than its LP counterpart because it attenuates the penny switching nature of linear models.

State-of-the-Art Modeling Technique: Positive Mathematical Programming

Positive Mathematical Programming (PMP) was originally introduced by Howitt (1995) as an efficient method for calibrating LP models and is now considered one of the mainstream methodologies employed for building farm

Box 1.

$$\begin{cases} \max\limits_{x^f} \quad gm^f\left(\mathbf{x}^f, \boldsymbol{\theta}^f, \boldsymbol{\kappa}\right) \equiv \sum\limits_{i=1}^{n}\left\{\left[\left(p_i \mid p_i^f + ps_i\right) y_i^f + ls_i - c_i^f\right] x_i^f\right\} \\ \text{s.t.} \quad \mathbf{A}^f\left(\boldsymbol{\theta}^f\right)\mathbf{x}^f \le \mathbf{b}^f\left(\boldsymbol{\theta}^f\right) \qquad \mathbf{A} \in \mathbb{R}^{m\times n} \\ \qquad\qquad \mathbf{x}^f \ge 0 \qquad\qquad \mathbf{x} \in \mathbb{R}^n \end{cases}$$

Table 2. Time lapse for parametric optimization of the regional model

	Solver	Time Elapsed in seconds		
Number of Steps		20	80	160
LP	CPLEX	7	30	61
NLP	Conopt3	13	34	66
PMP	Conopt3	34	66	118

Notes:
1. Problems solved @ Intel Pentium 4 CPU, 2,66 GHz with 0,5 GB of RAM
2. All solvers with default options

activity models. The PMP methodology is based on the assumption that the economic agent's observed behavior is optimal, because production choices rely not only on the observed parameter set appearing in the objective function, but also on additional implicit information that cannot be observed when examining farm data. This information is revealed by initially solving an LP model that exactly reproduces base year activity levels via a set of calibration constraints and is used to specify an onlinear, alternative objective function that allows for a model that calibrates. PMP specifications are expected to result in a perfect calibration and a zero AAD index, since their mathematical structure requires that the first order conditions of the underlying optimization problem are exactly satisfied when $x_i^{model} = x_i^{obs}$.

The PMP algorithm usually involves three phases during which the linear objective function is gradually transformed into a nonlinear one. The nonlinearity is usually sought in the cost term and is introduced in the model by replacing the linear cost function with a quadratic one, written in vector form as: $\mathrm{VC}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{d} + 0.5\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}$

Where $\mathbf{x}$ represents the unknown $n \times 1$ vector of activity levels, $\mathbf{d}$ is an $n \times 1$ vector of linear terms and $\mathbf{Q}$ is an $n \times n$ positive, semi-definite matrix that is either diagonal or fully specified. As mentioned above, the first phase of PMP involves using a simple LP model with additional calibration constraints that bind all activities at the observed level, $\hat{\mathbf{x}}$, thus forcing the model to exactly reproduce base year observations (see Box 2).

The last set of constraints binds each activity to its observed level and $\boldsymbol{\lambda}$ is the corresponding $n \times 1$ dual vector. Symbol ε denotes a small perturbation term that is used to prevent model degeneration caused by linear dependency among calibration and resource constraints. It is argued that $\boldsymbol{\lambda}$ embodies any type of marginal implicit information, such as model misspecifications, data errors, price expectations and farmer's risk attitude (Heck-

Box 2.

$$\begin{cases} \max\limits_{x^f} \quad gm^f\left(\mathbf{x}^f, \boldsymbol{\theta}^f, \boldsymbol{\kappa}\right) \equiv \sum\limits_{i=1}^{n}\left\{\left[\left(p_i \mid p_i^f + ps_i\right)y_i^f + ls_i - c_i^f\right]x_i^f\right\} \\ \text{s.t.} \quad \mathbf{A}^f\left(\boldsymbol{\theta}^f, \boldsymbol{\kappa}\right)\mathbf{x}^f \le \mathbf{b}^f\left(\boldsymbol{\theta}^f, \boldsymbol{\kappa}\right) \qquad \mathbf{A} \in \mathbb{R}^{m \times n} \\ \qquad\qquad \mathbf{x}^f \ge 0 \qquad\qquad \mathbf{x} \in \mathbb{R}^n \\ \qquad\qquad \mathbf{x}^f \le \hat{\mathbf{x}}^f + \varepsilon \qquad [\boldsymbol{\lambda}] \qquad \mathbf{x} \in \mathbb{R}^n \end{cases}$$

elei & Wolff, 2003). In this context, Paris and Howitt (1998) interpret $\boldsymbol{\lambda}$ as a "differential" marginal cost vector that, together with the observed "accounting" variable average cost, $\mathbf{c}$, reveals the actual variable marginal cost of the activities (**MVC**) at the observed production level. Hence, the derivative of the quadratic cost function at $\hat{\mathbf{x}}$ should be equal to $\boldsymbol{\lambda} + \mathbf{c}$:

$$\mathrm{MVC}(\hat{\mathbf{x}}) = \frac{\partial \mathrm{VC}}{\partial \mathbf{x}}(\hat{\mathbf{x}}) = \mathbf{d} + \mathbf{Q}\hat{\mathbf{x}} \quad (1)$$

The estimation of the quadratic cost function is based on equation (1) that constitutes an underdetermined system with n equations and $2n$ or $n + n(n+1)/2$ unknown parameters, depending on the form of the $\mathbf{Q}$ matrix; If $\mathbf{Q}$ is diagonal, several *ad hoc* methods have been proposed, summarized by Petsakos and Rozakis (2009), while for a fully specified matrix, Paris and Howitt (1998) propose the use of the maximum entropy criterion. This involves finding a discrete probability distribution over a vector of support values in order to maximize the entropy of the system and consequently the uncertainty on the value of the unknown parameters. In either case, the third phase of PMP consists in replacing the linear cost function with the estimated quadratic one and the final optimization model exactly reproduces base year observations.

For simplicity reasons we assume that $\mathbf{Q}$ is diagonal and for its estimation we follow the "average cost" approach, proposed by Heckelei and Britz (2000), according to which, the observed average variable cost, $\mathbf{c}$, should be equal to the average variable cost derived from the quadratic cost function (**AVC**):

$$\mathrm{AVC}(\hat{\mathbf{x}}) = \frac{\mathrm{VC}}{\mathbf{x}}(\hat{\mathbf{x}}) = \mathbf{d} + \frac{1}{2}\mathbf{Q}\hat{\mathbf{x}} = \mathbf{c} \quad (2)$$

Equations (1) and (2) define a system of $2n$ equations and $2n$ unknowns, namely the d_i elements of $\mathbf{d}$ and the q_i diagonal elements of $\mathbf{Q}$ (with $i = 1, 2, \ldots, n$), which finally yield:

$$\mathbf{Q} = \frac{2\boldsymbol{\lambda}}{\hat{\mathbf{x}}}$$

and

$$\mathbf{d} = \mathbf{c} - \boldsymbol{\lambda}$$

CASE STUDY

A web based Spatial Decision Support System has been implemented in Thessaly, the largest arable cropping region in Greece, in order to evaluate selected energy crop supply. To this end we use three alternative optimization models, namely an LP, an NLP and a PMP one. The methodology and architecture of this tool are detailed by Rozakis (2010). Energy to biomass raw material cost is provided in supply curve form incorporating physical land suitability for crops (survey and spatial information), farm structure (survey) and Common Agricultural Policy (CAP) scenarios. The analysis is based on a sample of 336 farms in the region of Thessaly that are included in the Greek Farm Accounting Data Network (FADN) for the year 2002. Each of these farms cultivates at least 0.1 hectares of either cotton or sugar beet at the same year. The estimates of variable costs per crop and farm mostly rely on the micro-economic data published by FADN, combined with survey data.

A set of farm specific "policy constraints" for cotton and tobacco was included for the analysis. More specifically, both cotton and tobacco areas for the previous CAP regime are constrained to the cultivated areas observed in the base year. For cotton, this reflects national policy to attenuate co-responsibility charges for exceeding a maximum guaranteed national quantity, while for tobacco it simulates the farm specific quotas that were active during the same period. The "resource constraints" used in both scenarios concern the availability of land and water. The constraint for total farm land was defined as an equation and not as a weak inequality, allowing the replacement of the constraint slackness with the "idle land" activity, in order to impose the cross compliance obligation of maintaining idle hectares in

"good agricultural condition". Water resources were modeled in terms of both irrigated area and total water quantity. For the former, a constraint bounding total irrigated land in each farm at the observed levels for the base year was used. For the latter, personal communication with experts provided information about average water requirements per hectare for each crop, which allowed the formulation of a constraint bounding the total water quantity in every farm to its 2002 estimated level. Both LP and NLP models for arable farms in Thessaly are specified in detail in Kampas et al. (2010). Each farm model contains in average 10 variables and 10 constraints, whereas the NLP specification includes only one quadratic term corresponding to alfalfa, its price determined in regional market places.

The LP model estimates satisfactorily land allocated to cotton, durum wheat and tobacco, as is shown in Table 1. In contrast, calibration fails for all other crops, especially soft wheat that is estimated at about 7.5 times the observed area. The same applies for the NLP specification, with the alfalfa inverse demand function, although it performs slightly better in reproducing base year situation, resulting in an average deviation of 1370.8 hectares (AAD) in land coverage for each crop comparing with 1799.1 for the LP model. As expected, PMP achieves perfect calibration, resulting in an AAD index value of zero.

Price to pay: computational complexity and solution time span gradually increases as presented in Table 2. One can observe that NLP solution time is about 20% greater than LP while PMP solution time is almost double the NLP time. As explained above response functions for output or other variables are implicit in mathematical programming models. They can be numerically determined by means of conducting several solutions of the model under parametric variation of the parameter in question. On this track, in order to generate biomass supply curves the optimization problem is parametrically solved for a number of steps within a price range determined by the user. Parametric optimization is naturally expected to swell differences in solution time span thus further penalizing the more advanced and demanding methods namely, NLP and PMP.

PARALLELIZATION OF THE PROBLEM

This problem constitutes an embarrassing parallelizable Mathematical Programming problem (EPMPP) since it is comprised of numerous independent problems. The elementary MP problem differs from any other only for a single coefficient in the objective function that is the energy crop price. EPMPP are excellent candidates for migrating to parallel solving because they can demonstrate very good speedup ratios.

Designing and solving mathematical programming problems can be described by the following "modeling life-cycle" (Fourer, 1996):

1. Defining model components. These are decision variables, objective function, the constraints and any related data. The first three above elements actually define the "model logic."
2. Solving model instances. Depending on the model type a different solver might be used. That means that the model instance should be expressed as a valid input for the appropriate solver.
3. Receiving results and interpreting them.

The above modeling life-cycle is usually implemented either in a high level programming language like FORTRAN and C or in a modeling environment like GAMS or AMPL. The former approach, while computationally more effective, it is much more complex in terms of implementation, requires special programming skills and is "overspecialized" to the narrow family of problems it addresses.

There is also the alternative of "mathematical programming notation" embedded in a high level programming language like Java (OptimJ) or Python (Coopr) or even in MATLAB (YALMIP). This approach provides flexibility without the disadvantages of modeling directly

to a programming language. This approach is of great interest to solving EPMPP since parallelization features are already present to all of the above programming languages.

But since the kind of MP problems we are interested are implemented in a modeling environment, we will focus on this latter.

Usually a modeling environment distinguishes between the phase of defining model components and the phase of solving its instances. At this last phase the appropriate solver is called and finally the solver updates the modeling environment with the solution details.

Based on the workflow, from the user perspective, parallelizing an EPMPP that is implemented in a modeling environment should have the following properties: Firstly, it would be ideal for the user that a parallel solution of his model would be initiated, executed and solved within the same modeling environment. Secondly there should be minor or even no changes to the user's model code. Also any setup of the parallel environment (like the number of nodes) would require either no user intervention or a very user friendly interface, within the modeling environment.

Some grid infrastructure issues that should be addressed include:

1. Network type, Intranet or the Internet? Cloud technologies, like Amazon EC2, are using the Internet while simple cluster and HPC solutions are based on intranets. Napper (Napper, 2009) has run the LINPACK benchmark to an Amazon EC2 cloud and has found that "the cost for solving a linear system increases exponentially." Cloud computing is evolving and whether it can replace HPC scientific computing is an ongoing issue (Ramakrishnan, 2011; Zhai, 2011; Vöckler, 2011; Brandic & Raicu, 2011).
2. Intranet Network Technology: Different network technologies (like ethernet, myrinet, infiband) perform differently in terms of latency and also have different installation costs. Ethernet technology is the most common in everyday academic life while Myrinet and Infiband is usually present in HPC environments as being more effective. For EPMPP we expect the computation time per problem to be much longer than communication time, so an Ethernet solution is expected to be adequate.
3. Dedication of Infrastructure to the solution of distributed problems: When the distributed environment is dedicated to solving parallel problems then we expect total computing performance to be higher (for example HPC systems). On the other hand such an environment might be difficult to be available and also it is costly to be build by small academic units. If such is the case an HTC system is a good alternative. HTC systems (like Condor) use workstations to harness their idle computing power. For EPMPP problems an HTC system is expected to be sufficient.
4. Installation and administration issues: Also it should be noted that although it is getting simpler and simpler to build and maintain a distributed computing environment, it requires specialized knowledge and skills. Even if a Distributed Computing Infrastructure is available, some issues that possibly would have to be addressed for solving EPMPP problem are: the need to install software (like the modeling environment or the required solvers or the web service) to all the workstations of the cluster, Provision for the possibility that a client might fail to return results.

CURRENT IMPLEMENTATIONS FOR OPTIMIZATION IN PARALLEL

An early paper on the issue of distributed optimization services, and thus on the issue of solving EPMPP in parallel, is that "Optimization as an Internet Resource" (Fourer, 2000). It provides a list of websites that at that time provided optimization services in some way and describes efforts for utilizing internet to

Table 3. Results of survey for the capability of modeling environments to submit to solvers in a Grid

Modelling Environment	Vendor	Can submit instances to Grid
AIMMS	Paragon Decision Technology	No
AMPL	AMPL Optimization LLC	No
FICO Xpress Optimization Suite	FICO	n/a
GAMS	GAMS Development Corporation	Yes
GIPALS - Linear Programming Environment	Optimalon Software Ltd	No
IBM ILOG CPLEX Optimization Studio	IBM	n/a
LINGO	LINDO Systems, Inc.	No
Mathematical Modeling System (LPL)	Virtual Optima	No
Oracle Crystal Ball Suite	Oracle Americas Inc.	No
QMS	QuantMethods	No
SCIP (Zimpl)	Zuse Institute Berlin	No
Solver SDK Platform	Frontline Systems, Inc.	No
Vanguard Global Optimizer	Vanguard Software	Yes

Notes:

1. In the above list we include only modeling environments. No excel add-ins, no solvers, no "LP modeling notation" programming language extensions.
2. This list is a subset of the 2011 Linear Programming Software Survey of OR/MS TODAY
3. We include only the modeling environments that their vendors replied to our email
4. n/a is no answer

provide mathematical programming solver services with the technology of that time like NEOS server, iNEOS, AMPL remote access, MetaNEOS, iMW).

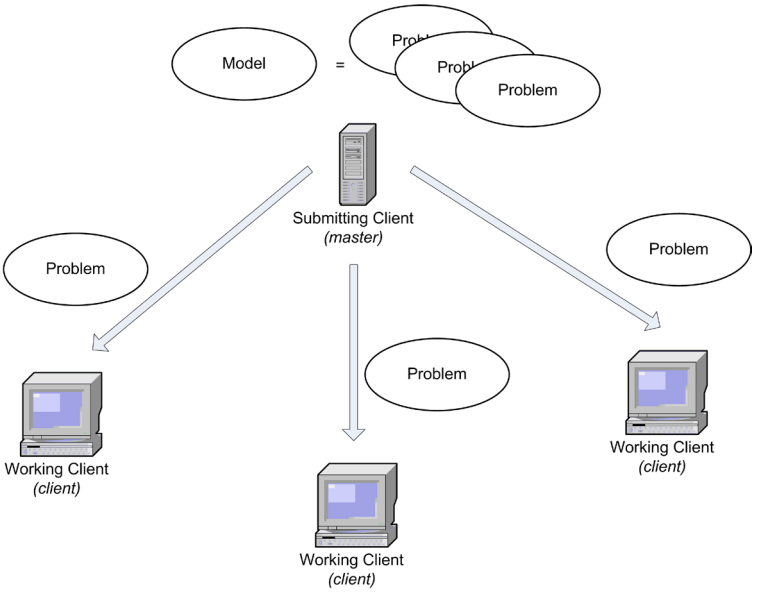
Using the list of "Linear Programming Survey" from "OR/MS Today", we have contacted vendors of Linear Programming software that falls into the category of "Modeling Environment" and we have compiled a table of whether or not they provide inline (within their modeling environments) capability for submitting model instances to a grid. In Table 3 we provide the results of our survey. Only two modeling environments (GAMS and Vanguard Global Optimizer) inherently possess the capability to submit model instances to a grid. Most of the other environments have APIs that could be used in order to craft a parallel solution that will submit model instances to nodes running the solvers.

GAMS griding facility provides a mechanism for submitting solves (that is model instances) asynchronously to a grid infrastructure, and housekeeping the ongoing solves and the results. Information for setting up GAMS griding facility can be found in Bussieck (2009) and Ferris (2000).

In order to use Vanguard Global Optimizer to solve optimization problems in parallel, one has to setup a Vanguard simulation grid and then submit its model instance to the central Vanguard server. Vanguard also provides a high performance computer grid for use as a paid subscription.

Although not a modeling environment by itself, Optimization Services (OS) is "a set of standards for representing optimization instances, results, solver options, and communication between clients and solvers in a distributed environment using Web Services". The importance of this effort for the cause of

Figure 1. Architecture of epLPpMS



solving EPMPP to a grid is obvious. A related discussion of design and implementation issues that arise for providing optimization as a service can be found at Fourer (2008). Also it should be noted that client and server interfaces are provided for download, so setting up a grid with Optimization Services is readily available.

NEOS Server should also be mentioned (Czyzyk et al., 1998) as it has the capability of sending model instances to remote solvers. Augmented with "Kestrel" (Dolan et al., 2008), an automated way of interacting with the server, it can be used for solving EPMPP in parallel.

AN AD HOC IMPLEMENTATION: EMBARRASSING PARALLELIZABLE LINEAR PROGRAMMING PROBLEMS META SOLVER (EPLPPMS)

Although our case study's model was expressed in GAMS code, we did not use GAMS griding facility as the appropriate script files for submitting to our type of grid was not available. So we have turned to a meta-solver approach (more accurately a meta-modeling-environment approach) developing epLPpMS. It is an ad hoc solution that needs almost no changes to the existing model code and can operate efficiently in a small cluster like a PC-Lab in the Academia. epLPpMS means "embarrassing parallelizable Linear Programming problems Meta Solver." It is a master-worker architecture application that is written in Java and aims at solving embarrassingly parallelizable LP or QP problems. epLPpMS is currently implemented for the GAMS modeling environment but due to the flexibility of the object oriented nature of JAVA programming language, there is a potential for extending the application for other modeling environment too. The description of how the model should be partitioned and what workstations are available as workers is given through an XML file. The worker program is installed as a web service. Executable code for both master and worker programs and installation instructions of epLPpMS software can be found at http://aoatools.aua.gr/_sites/epLPpMS

Initially the model is transformed to the elementary problems that will be solved in parallel. Then these problems are transmitted

Table 4. Parallel solving time and computation speedup, LP, NLP, and PMP models

		LP model		NLP model		PMP model	
price steps	no of pcs	time (sec)	speedup	time (sec)	speedup	time (sec)	speedup
	2	4.19	1.67	7.96	1.63	22.60	2.70
20	4	4.93	1.42	7.03	1.85	22.63	2.70
	8	5.01	1.40	8.11	1.60	30.77	1.98
	12	5.43	1.29	8.01	1.62	30.92	1.97
	16	5.68	1.23	8.75	1.48	30.58	1.99
	2	12.06	2.49	15.59	2.18	36.03	1.83
80	4	8.35	3.59	12.06	2.82	30.72	2.15
	8	7.35	4.08	10.82	3.14	25.32	2.61
	12	6.97	4.30	10.31	3.30	25.91	2.55
	16	7.51	3.99	7.89	4.31	29.78	2.22
	2	22.18	2.75	25.78	2.56	48.12	2.45
160	4	14.62	4.17	17.85	3.70	39.05	3.02
	8	10.51	5.80	14.75	4.48	35.29	3.34
	12	9.48	6.43	12.47	5.29	34.05	3.47
	16	7.90	7.72	11.86	5.57	32.16	3.67

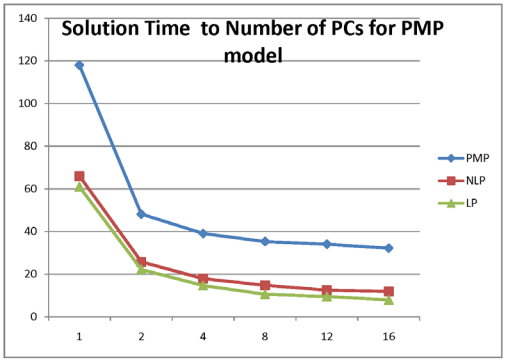
to the worker machines that send the results back after the process of optimization is completed. The user initiates the master process which is responsible for breaking the model in the multiple EPLP problems and also for their transmission to the remote worker processes. The worker processes are installed on the client machines located on the LAN and are responsible for solving the transmitted instances of the model. An overview of its architecture can be seen in Figure 1.

Although epLPpMS performed well for the needs of our case study, it is in an elementary stage of development and many improvements should be made: Making a simple web interface for submitting jobs to the grid and thus replacing the current command line interface. Inheriting a standard format for representing and exchanging models, solver options and results, like OSiL. Migrating to a well established grid infrastructure, like GridGain, Condor or even Amazon EC2, instead of our raw implementation. Finally, the model instances distribution algorithm should also be improved towards a more hierarchical distribution scheme in order to lighten the load to the node that submits all the model instances.

CASE STUDY RESULTS

After doing a minimal modification on GAMS model code and creating the appropriate xml input files we have run the web-DSS model in parallel. The operation took place at the Department of Agricultural Economics PC-Lab, where 20 windows workstations are currently in operation. Half of the PCs are Intel Pentium-4 2.3 MHz with 1Gb of RAM and the others are Intel Pentium Core-Duo 2.1 MHz with 512Gb of RAM. The network topology is Ethernet at 100Mbs. We have collected the time elapsed for solving in parallel the LP model and the Non-LP model for 20, 80,160 steps of a certain price range for energy crop at 2,4,8,12,16 PCs. We have run the test for each combination three times. The results are presented in Table 4. Also the graphs of the speedup (the ratio of the serial

Figure 2. Solution time speedup to number of PCs for NLP



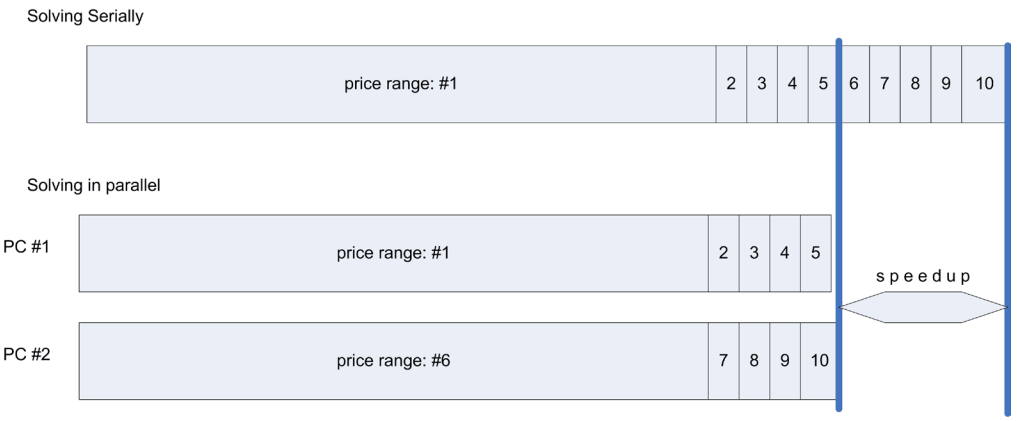
solution time to the parallel solution time) to the number of PCs are presented in Figure 2.

As we can see, for a small sized problem like 20 price steps the speedup is either a slow-down or insignificant. As the problem size is growing, like in the case of 160 price steps where the total time needed to solve serially the model exceeds a TWT, the speedup of the parallel solution becomes more important.

One could argue that PMP does not scale well, since for a problem size of 160 price steps, solving with 16 processors, the speedup ratio is only 3.67. This is explained by the fact that for a serial solution the GAMS/CONOPT solver spends over 90% of the total solution time to the first price step and only the time for the rest price steps can be shortened with parallel solving. For NLP, the parallel computing performance is better since the first price step solution time drops below 50% of the total time. The above is depicted in Figure 3 for parallel solving in 2 computers. It is clear that if solving times were equal for all price steps, then the speedup would converge to the number of computers that were participating.

The reason for this asymmetry in the solving times is due to the fact that the GAMS/CONOPT solver starts each solve from the previous solution and since the change to the model data for every price step is insignificant the solution time for the second and above solve is very small.

Figure 3. Explaining the "poor" speedup performance of PMP



CONCLUSION

Parallel computing is used to enhance the decision process quality regarding bio-energy projects evaluation. This is achieved thank to reduction in solution time of models that support decision making especially behavioral models that simulate farmers' response to prices signals emitted by the industry.

Improvement is most significant for many iterations and in the case of Non LP models that are computationally more demanding than equivalent LP problems. PMP and NLP problems are preferred over LP because they make the DSS tool more reliable, able to survive in a business environment, but at a price of higher computing time duration. We have proved that parallel computing can palliate this problem making the web-SDSS tool more user friendly.

Further research is needed to accommodate the parallel computing algorithm in order to test advanced alternative model specifications representing state-of-the-art of regional modeling techniques, taking into account risk and uncertainty in farmer behavior as well as positive approaches to agricultural supply modeling. Also several implementation and integration issues have to be addressed, like immediate distributed resource availability and the cost-benefit of the migration to a parallel solution. Finally, Optimization Services (Fourer, 2008) looks very promising for the cause of parallelizing embarrassingly parallelizable MP problems and it would be very interesting to compare our implementation with one that uses OS.

REFERENCES

Bauer, S., & Kasnakoglou, H. (1990). Non-linear programming models for sector and policy analysis. *Economic Modelling*, *7*(3), 275–290. doi:10.1016/0264-9993(90)90013-T

Bhargava, H. K., Power, D. J., & Sun, D. (2007). Progress in Web-based decision support technologies. *Decision Support Systems*, *43*, 1083–1095. doi:10.1016/j.dss.2005.07.002

Brandic, I., & Raicu, I. (Eds.). (2011). Science-driven cloud computing. *Scientific Programming, 19*(2-3).

Bussieck, M. R., Ferris, M. C., & Meeraus, A. (2009). Grid-enabled optimization with GAMS. *INFORMS Journal on Computing*, *21*, 349–362. doi:10.1287/ijoc.1090.0340

Czyzyk, J., Mesnier, M., & Moré, J. (1998). The NEOS Server. *IEEE Journal on Computational Science and Engineering*, *5*, 68–75. doi:10.1109/99.714603

Dolan, E. D., Fourer, R., Goux, J.-P., Munson, T. S., & Sarich, J. (2008). Kestrel: An Interface from optimization modeling systems to the NEOS Server. *INFORMS Journal on Computing*, *20*, 525–538. doi:10.1287/ijoc.1080.0264

Ferris, M. C., & Munson, T. S. (2000). Modeling languages and Condor: Metacomputing for optimization. *Mathematical Programming*, *88*, 487–506. doi:10.1007/PL00011382

Fourer, R. (1996). Software for optimization: A buyer's guide. Part 1. *INFORMS Computer Science Technical Section Newsletter, 17*(1), 14-17. Retrieved September 1, 2011, from http://www.ampl.com/buyers1.html

Fourer, R., & Goux, J. P. (2001). Optimization as an Internet Resource. *Interfaces*, *31*(2), 130–150. doi:10.1287/inte.31.2.130.10629

Fourer, R., Ma, J., & Martin, K. (2008). Optimization services: A framework for distributed optimization. *Operations Research*, *58*, 1624–1636. doi:10.1287/opre.1100.0880

Fourer, R., Ma, J., & Martin, K. (2010). OSiL: An instance language for optimization. *Computational Optimization and Applications*, *45*(1), 181–203. doi:10.1007/s10589-008-9169-6

Goffe, W. L., & Creel, M. (2005). *Multi-core CPUs, clusters and grid computing: A tutorial* (Computing in Economics and Finance 2005, No. 438). Society for Computational Economics. Retrieved September 1, 2011, from http://ideas.repec.org/p/sce/scecf5/438.html

He, Q., Zhou, S., Kobler, B., Duffy, D., & McGlynn, T. (2010). Case study for running HPC applications in public clouds. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (pp. 395-401). New York, NY: ACM.

Heckelei, T., & Britz, W. (2000). Positive mathematical programming with multiple data points: A cross-sectional estimation procedure. *Cahiers d'Economie et Sociologie Rurales*, *57*, 28–50.

Heckelei, T., & Wolff, H. (2003). Estimation of constrained optimisation models for agricultural supply analysis based on generalized maximum entropy. *European Review of Agriculture Economics*, *30*, 27–50. doi:10.1093/erae/30.1.27

Howitt, R. E. (1995). Positive mathematical programming. *American Journal of Agricultural Economics*, *77*(2), 329–342. doi:10.2307/1243543

Institute for Operations Research and the Management Sciences. (2011). *Linear programming software survey*. Retrieved from http://www.orms-today.org/surveys/LP/LP-survey.html

Kampas, A., Petsakos, A., & Rozakis, S. (2010). *Price induced water irrigation: Unraveling conflicts and synergies between European agricultural and water policies* (Working Paper No. 2010-05). Athens, Greece: Agricultural University of Athens.

Kingston, R., Carver, S., Evans, A., & Turton, I. (2000). Web-based public participation geographical information systems: an aid to local environmental decision-making. *Computers, Environment and Urban Systems*, *24*, 109–125. doi:10.1016/S0198-9715(99)00049-6

Kutcher, G. P., & Norton, R. D. (1979). Operations research methods in agricultural policy analysis. *European Journal of Operational Research*, *10*, 333–345. doi:10.1016/0377-2217(82)90084-4

Manos, B., Bournaris, T., Papathanasiou, J., & Chatzinikolaou, P. (2009). Evaluation of tobacco cultivation alternatives under the EU CAP. *Journal of Policy Modeling*, *31*, 225–238. doi:10.1016/j.jpolmod.2008.07.001

McKone, T. E., Nazaroff, W. W., Berck, P., Auffhammer, M., Lipman, T., & Torn, M. S. (2011). Grand challenges for life-cycle assessment of biofuels. *Environmental Science & Technology*, *45*, 1751–1756. doi:10.1021/es103579c

Nah, F. F.-H. (2004). A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, *23*(3), 153–163. doi:10.1080/01449290410001669914

Napper, J., & Bientinesi, P. (2009). Can cloud computing reach the top500? In *Proceedings of the Combined Workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop* (pp. 17-20). New York, NY: ACM.

Paris, Q., & Howitt, R. E. (1998). An analysis of ill-posed production problems using maximum entropy. *American Journal of Agricultural Economics*, *80*, 124–138. doi:10.2307/3180275

Petsakos, A., & Rozakis, S. (2009). Critical review and state-of-the-art of PMP models: an application to Greek arable agriculture. In A. Rezitis (Ed.), *Research topics in agricultural and applied economics* (Vol. 1, pp. 36-61). Sharjah, UAE: Bentham Science.

Power, D. J. (2004). Specifying an expanded framework for classifying and describing decision support systems. *Communications of the Association for Information Systems*, *13*(13), 158–166.

Ramakrishnan, L., Zbiegiel, P. T., Campbell, S., Bradshaw, R., Canon, R. S., & Coghlan, S. …Liu, A. (2011). Magellan: experiences from a science cloud. In *Proceedings of the 2nd international workshop on Scientific cloud computing* (pp. 49-58). New York, NY: ACM.

Rozakis, S. (2010). A Web-based Spatial DSS for estimating biomass-to-energy supply in Thessaly . In Manos, B., Paparrizos, K., Matsatsinis, N., & Papathanasiou, J. (Eds.), *Decision support systems in agriculture, food and the environment* (pp. 450–465). Hershey, PA: IGI Global. doi:10.4018/978-1-61520-881-4.ch021

Rozakis, S. (2011). Impacts of flatter rates and environmental top-ups in Greece: A novel mathematical modeling approach. *Agricultural Economics Review, 12*(2).

Rozakis, S., Tsiboukas, K., & Korasidis, M. (2008). CAP reform impacts to Greek cotton farmers: A mathematical programming approach. *Journal of Agricultural and Food Economics*, *3*(1-2), 85–106.

Simon, D., Tyner, W. E., & Jacquet, F. (2010). Economic analysis of the potential of cellulosic biomass available in France from agricultural residue and energy crops. *Bioenergy Research*, *3*(2), 183–193. doi:10.1007/s12155-009-9061-y

Sugumaran, V., & Sugumaran, R. (2007). Web-based Spatial Decision Support Systems (WebSDSS): Evolution, architecture, and challenges. *Communications of the Association for Information Systems, 17*(1), 844-875. Retrieved September 1, 2011, from http://aisel.aisnet.org/cais/vol19/iss1/40

Van der Hilst, F., Dornburg, V., Sanders, J. P. M., Elbersen, B., Graves, A., & Turkenburg, W. C. (2010). Potential, spatial distribution and economic performance of regional biomass chains: The North of the Netherlands as example. *Agricultural Systems*, *103*(7), 403–417. doi:10.1016/j.agsy.2010.03.010

Vöckler, J. S., Juve, G., Deelman, E., Rynge, M., & Berriman, B. G. (2011). Experiences using cloud computing for a scientific workflow application. In *Proceedings of the 2nd international workshop on Scientific cloud computing* (pp. 15-24). New York, NY: ACM.

Zhai, Y., Liu, M., Zhai, J., Ma, X., & Chen, W. (2011, November). Cloud versus in-house cluster: evaluating Amazon cluster compute instances for running MPI applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-10). Washington, DC: IEEE Computer Society.