HAL open science

Combined Aggregation and Column Generation for Land-Use Trade-Off Optimisation

Asef Nazari, Andreas Ernst, Simon Dunstall, Brett Bryan, Jeff Connor, Martin Nolan, Florian Stock

▶ **To cite this version:**

Asef Nazari, Andreas Ernst, Simon Dunstall, Brett Bryan, Jeff Connor, et al.. Combined Aggregation and Column Generation for Land-Use Trade-Off Optimisation. 11th International Symposium on Environmental Software Systems (ISESS), Mar 2015, Melbourne, Australia. pp.455-466, 10.1007/978-3-319-15994-2_46 . hal-01328592

HAL Id: hal-01328592

https://inria.hal.science/hal-01328592v1

Submitted on 8 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Combined Aggregation and Column Generation for Land-Use Trade-Off Optimisation

Asef Nazari[1], Andreas Ernst[1], Simon Dunstall[1], Brett Bryan[2], Jeff Connor[2], Martin Nolan[2], Florian Stock[3]

[1]CSIRO: Digital Productivity and Services, Melbourne, Australia {asef.nazari, andreas.ernst, simon.dunstall}@csiro.au
[2]CSIRO: Land and Water, Adelaide, Australia {brett.bryan, jeff.connor, martin.nolan}@csiro.au
[3]CSIRO: Digital Productivity and Services, Melbourne, Australia, and InnoZ – Innovation Centre for Mobility and Societal Change, Berlin, Germany florian.stock@gmex.de

Abstract. In this paper we developed a combination of aggregation-disaggregation technique with the concept of column generation to solve a large scale LP problem originating from land use management in the Australian agricultural sector. The problem is to optimally allocate the most profitable land use activities including agriculture, carbon sequestration, environmental planting, bio-fuel, bio-energy, etc., and is constrained to satisfy some food demand considerations and expansion policies for each year from 2013 to 2050. In this research we produce a higher resolution solution by dividing Australia's agricultural areas into square kilometer cells, which leads to more than thirteen million cells to be assigned, totally or partially, to different activities. By accepting a scenario on agricultural products' return, carbon related activities, future energy prices, water availability, global climate change, etc. a linear programming problem is composed for each year. However, even by using a state of the art commercial LP solver it takes a long time to find an optimal solution for one year. Therefore, it is almost impossible to think about simultaneous scenarios to be incorporated, as the corresponding model will become even larger. Based on the properties of the problem, such as similar economical and geographical properties of nearby land parcels, the combination of clustering ideas with column generation to decompose the large problem into smaller sub-problems yields a computationally efficient algorithm for the large scale problem.

Keywords: aggregation-disaggregation · column generation · clustering · land use allocation

1 Introduction

Predicting land use change in Australian agriculture in the context of increasing energy prices stimulating bio-fuels and bio-energy land uses, and a carbon policy with possibilities for increasing carbon price over time requires solving a large scale linear programming problem. The model covers the domain of southern and eastern Austral-

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

ian land currently in intensive agricultural use on a one square kilometer grid cell resolution over 813,000 square kilometers resulting in more than thirteen million square cells, including active and inactive cells, as shown in Fig. 1. Starting with the agricultural land present in 2013, the model finds the optimal land use each year forward in time until 2050 where optimality is defined as maximizing profit and social welfare (the sum of profit and consumers' surplus). In the highest resolution with land parcels of size one square kilometer, for each year the corresponding LP has 7,313,847 continuous variables and 814,811 constraints. The focus is on the change from current agricultural production to alternative land uses such as carbon plantings, environmental plantings, bio-energy or bio-fuels. Food prices are computed endogenously following a maximum welfare approach whereas costs and revenues for non-agricultural commodities depend on scenarios and exogenous modeling. The platform is built to model a range of scenarios involving alternate assumptions about global climate change, world carbon emissions trajectories, emissions limits and prices of carbon credits, price trajectories for energy, world food demand, supply and price trajectories, and agricultural productivity growth [11].

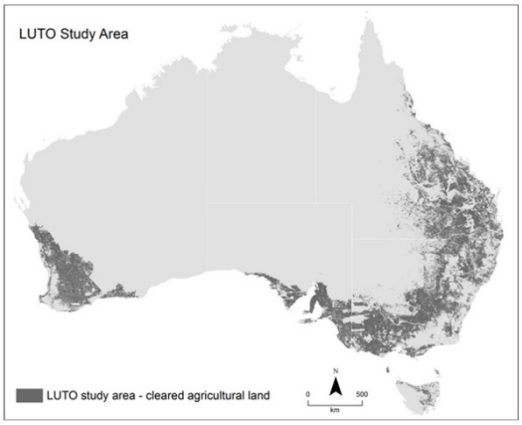

Fig. 1. Area for land use trade optimisation

Efficiently solving a large scale LP problem is the main difficulty that we address in this paper. At the present setting, finding an optimal solution for each year takes a long time even using commercial software packages. Also, considering that the model is build based on one set of fixed scenarios, the further expansion of the project to consider simultaneous scenarios for the sake of stochastic optimisation is almost impossible at this stage. Therefore, our first aim is to develop an algorithm which can solve the large LP pertaining to a particular year in a shorter amount of time.

One important fact in terms of agricultural land allocation for different products is that nearby land parcels have similar economical and geographical properties. They indeed have similar proximity to water resources, and main roads leading to similar transportation and production costs. In reality it is easy to detect that huge parcels of land are allocated to the same agricultural commodity while passing large farms in

countryside. By drawing inspiration from this fact, instead of considering cells of size one square kilometer, the whole Australia is divided into big chunks of land (lower resolution) called clusters. The economical properties of each cluster are calculated based upon cells in each cluster. This type of aggregation technique in optimisation is a handy tool to create a set of smaller problems out of a large problem. However, the smaller problem size comes at the cost of lower accuracy. We remedy this by using an iterative disaggregation approach. The smaller problem is gradually made a better approximation of the original problem by breaking down big clusters into smaller clusters, and add new clusters as new column to the small problem.

There is an increasing level of attention to use optimisation techniques in agricultural and other land use management planning context. As an example, in [7] a hierarchical approach is presented for large-scale forest planning. The algorithm is based on solving an aggregate problem, which is of moderate size. Another example, in [8] authors argue the usage of optimisation techniques in combination with scenario analysis can provide efficient land use management options for sustainable land use from global to sub-global scales. In terms of water resource management, in [9] a spatial optimisation techniques implemented among four diffuse source pathways in a mixed-use watershed to maximize total reduction of phosphorus loading to streams while minimizing associated costs. An interesting utilization of a multi-objective optimisation technique is reported for identifying optimum land management adaptations to climate change [10].

The rest of the paper is arranged as follows. In the next sections, we describe the model in more details. Then a short description of column generation and aggregation techniques is provided. The new algorithm is introduced at the next section, and numerical results and the conclusion will conclude the paper.

2 The Model

In order to insert demand-production equilibrium, a segmentation of the domain is necessary to approximate the nonlinear relationship between the demand and production using piecewise linear functions. Also, an appropriate model needs to consider satisfaction of the Australian agricultural food demand by imposing supply-production and supply-demand constraints. As we should allocate every cell, partially or totally, to the activities, a set of land-use constraints are introduced. Furthermore, in case there are some particular expansion plans for particular activities, some new expansion constraints are introduced to fulfill existing capacity for each activity. The complete model of the land-use trade-off optimization (LUTO) project is represented by equation (1). Some necessary and technical information of the model is represented in Table 1.

With resolution set to one, which is the highest resolution with land parcels of size one km^2, there are $|R|$=812,383 regional active cells. Also, there exist $|S|$=100 price and demand segmentation, $|J|$=9 activities and $|F|$=24 food commodities. The linear model has $|R|*|J|+|S|*|F| \approx$ 7,313,847 continuous variables and $|N|+|R|+|F|+|S|*|F| \approx$

814,811 constraints. The number of variables is almost nine times more than the number of constraints.

$$
\begin{cases}
\max \sum_{j \in F} \sum_{s \in S} P_{sj} y_{sj} - \sum_{j \in F} \sum_{r \in R} C_{r,ag_j} Q_{r,ag_j} x_{r,ag} + \sum_{j \in N} \sum_{r \in R} \delta_{rj} Q_{rj} x_{rj} \\
s.t. \quad \sum_{r \in R} Q_{rj} x_{rj} \leq Lim_j \quad j \in \{\text{carbon, biofule, bioenergy}\} \subset N \quad \text{Expansion} \\
\sum_{r \in R} Q_{r,epB} x_{r,epB} \leq Lim_{epB} \qquad \text{Biodiversity} \\
\sum_{j \in J} x_{rj} \leq 1 \qquad \forall r \in R \qquad \text{Land-Use} \\
\sum_{s \in S} y_{sj} \leq \sum_{r \in R} Q_{r,ag_j} x_{r,ag} \qquad \forall j \in F \qquad Supply - Production \\
y_{sj} \leq D_{sj} \quad \forall s \in S, \quad j \in F \qquad Supply - demand \\
x_{rj}, y_{sj} \geq 0 \qquad \forall s \in S, r \in R, i \in F, j \in J
\end{cases}
\tag{1}
$$

Sets	
J	Set of 9 activities including agriculture, carbon sequestration, environmental planting, biodiversity, tree-based bioenergy, wheat-based bioenergy, wheat-based biofuel, tree-based biofuel and wheat-based biofuel and food represented as $\{Ag, Cp, Ep, EpB, BeWP, BeS, \text{BfGS}, \text{BfWP}, \text{BfFS}\}$.
F	Set of 24 food commodities, ag_j's.
N	Non-food activities including Carbon, bioenergy, biofuel and environmental planting activities; In other words activities in J other than Ag.
R	Set of regions.
S	Set of segments in discrete food demand function.
Variables	
x_{rj}	Ratio allocated to activity $j \in J$ at cell $r \in R$ $(0 \leq x_{rj} \leq 1)$,
y_{sj}	Amount of commodity $j \in F$ should be produced at the segment $s \in S$ at price P_{sj} .
Constants	
Q_{sj}	Quantity of activity j that can be produced at cell r.
δ_{rj}	Annualized economic return of activity j at r.
D_{sj}	Demand for commodity j at segment s, $\sum_s D_{sj} = D_j$.
C_{rj}	Production cost of commodity j at region r.
P_{sj}	Price of commodity j in segment s at demand level y_{sj}.

Table 1. Components of the model

3 Column Generation

Column generation is a widely used technique to solve large scale linear and integer programming problems starting from pioneering publications [1] and [2]. The technique is frequently utilized when the number of variables is much larger that the number of constraints. In such large scale problems, the vast majority of the variables are zero at optimality, hence the fundamental concept underlying column generation is to solve a smaller problem instead of the original LP by considering a subset of columns (variables). The smaller problem is referred as restricted master problem (RMP), and new columns are added as required. The generation of new columns is accomplished by solving another problem called pricing sub-problem following each optimisation of the RMP. In the pricing stage of each iteration, the column(s) with minimum (maximum, depending on the objective) reduced cost(s) are added to the RMP. The optimality is achieved if it is impossible to add a new column to the RMP. For comprehensive surveys on column generation interested readers can consult [3] and [4].

For a formal explanation of the column generation algorithm consider the linear programming (1) where the $\boldsymbol{p}_j$ is the jth column of the coefficient matrix, and the number of variables is extremely greater than the number of constraints. Assume that an initial basic feasible solution, $\boldsymbol{x}_B$, is available, with associated basis matrix B, and cost coefficient $\boldsymbol{c}_B$. The simplex multipliers associated with this basis could be calculated as $\pi = \boldsymbol{c}_B B^{-1}$ and are always available at each iteration of the simplex algorithm. To improve the basic feasible solution we "price out" all columns corresponding to non-basic variables by forming their corresponding reduced cost $\overline{c_j} = c_j - \pi \boldsymbol{p}_j$. If $\max \overline{c_j} = \overline{c_s} > 0$, then considering non-degeneracy, the current solution may be improved by introducing x_s into the basis via a pivot transformation.

4 Aggregation Techniques

An important issue in obtaining an optimal solution of large scale optimisation problems is the trade-off between the level of details and the ease of solving the model in an acceptable amount of time. Aggregation-disaggregation techniques provide some methodologies for handling large optimisation problems by combining data, or using aggregated problems which are reduced in size. One main approach to construct an aggregated problem for a large scale LP is by partitioning the variables and forming corresponding columns by weighted average of columns in each partition. A feasible solution of the original problem is then achieved by applying a special transformation, called disaggregation, to an optimal solution of the aggregated problem. The aggregation error calculated as the difference between the original optimal objective value and the optimal value for the disaggregated solution guides the algorithm towards the optimality in an iterative scheme. In this section, a concise description of aggregation inspired from [5] is provided.

Consider the original linear programming problem in the form of

$$(LP)\begin{cases} z^* = \max \boldsymbol{cx} \\ s.t.\ A\boldsymbol{x} \le \boldsymbol{b}\ (2) \\ \boldsymbol{x} \ge \boldsymbol{0} \end{cases}$$

A column aggregation is explained here, which means only variables in (2) are aggregated. Let σ be a partition of the column indices $\{1, \dots, n\}$ into a set of ters$S_k, k = 1, \dots, K$, such that$S_k \cap S_p = \emptyset$, $\bigcup_{k=1}^{K} S_k = \{1, \dots, n\}$. For the kth cluster, let its size as$|S_k| = n_k$, so that $\sum_k n_k = n$. The matrix A^k is defined to be the sub-matrix of A consisting of those columns whose indices are in S_k. Also, the subvectors $\boldsymbol{c}^k$ and $\boldsymbol{x}^k$ are defined in the same way. Consider a nonnegative n_k-vector $\boldsymbol{g}^k = (g_j^k)$, satisfying the following normalizing condition

$$\boldsymbol{g} \in G = \{\boldsymbol{g} | \sum_{j \in S_k} g_j^k = 1,\ \ g_j^k \ge 0,\ \ k = 1, \dots, K, j \in S_k\}$$

and form

$$\overline{A^k} = A^k \boldsymbol{g}^k, \qquad \overline{\boldsymbol{c}^k} = \boldsymbol{c}^k \boldsymbol{g}^k, \qquad k = 1, \dots, K$$

such that $\overline{A^k}$ is a column m-vector equal to the linear combination of the columns $\boldsymbol{a}_j, j \in S_k$ with the coefficients g_j^k , and $\overline{\boldsymbol{c}^k}$ is a scalar defined similarly. The vectors $\boldsymbol{g}^k$ are called weighting vectors or the weights of aggregation.

Define the matrix $\bar{A} = [\overline{A^1}, \dots, \overline{A^K}] = [\overline{a_{ik}}]$, and the vector $\bar{\boldsymbol{c}} = (\overline{\boldsymbol{c}^1}, \dots, \overline{\boldsymbol{c}^k})$. Then the problem

$$(ALP)\begin{cases} \bar{z} = \max \bar{\boldsymbol{c}}\boldsymbol{X} \\ s.t.\ \bar{A}\boldsymbol{X} \le \boldsymbol{b}\ (3) \\ \boldsymbol{X} \ge \boldsymbol{0} \end{cases}$$

defines the (column or variable) aggregated problem corresponding to (2). In (3), $\boldsymbol{X}$ is a K-vector of aggregated variables. For a given original problem, then, an aggregated problem is determined by the pair $(\sigma, \boldsymbol{g})$. It is assumed that $(\sigma, \boldsymbol{g})$ has been chosen so that (3) is feasible. The optimal primal and dual solutions of the aggregated problem are denoted as $(\boldsymbol{X}^*, \pi^*)$. Interested readers could consult [6] for a general framework for aggregation and disaggregation technique and a survey on previous works.

5 The New Approach

Solving LP (1) takes a long time even using state of the art LP solver CPLEX as depicted in Fig. 2. For a particular instance of LP (1) used to sketch the chart, finding an optimal solution of the LP takes maximum time of 1:08:34, minimum amount of 0:21:34, and in average it takes 36 minutes per year for a complete run for a period of 38 years. Considering that the model should be solved for several consecutive years, at least for 38 years in this project, obtaining the optimal land allocation policy for a long planning horizon becomes quite time consuming. In addition, all the economic data, demand and supply figures along with energy price paths are acquired based on only one scenario out of thousands of scenarios. As a consequence, considering more

than one scenario simultaneously for the sake of a stochastic model will result in an even larger problem and will require longer computational time. Therefore, for more comprehensive investigations on optimal land allocations an efficient solution methodology is necessary to solve the large LP problem in a reasonably shorter amount of CPU time.

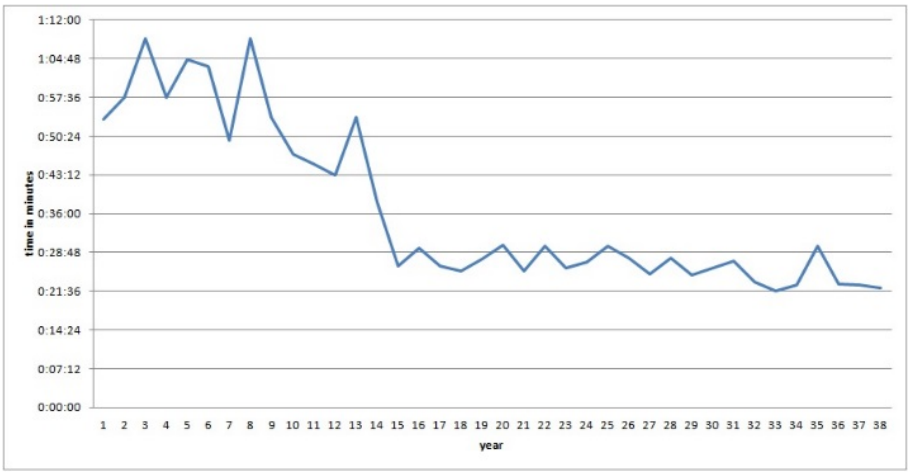

Fig. 2. CPU time for solving the LP for each year from 2013 to 2050

There are some interesting properties associated with agricultural land allocation that are utilized in our aggregation process. For example, it is quite plausible that nearby land parcels would be allocated to similar activities. The reason for this assumption is that geographically close areas have similar economical and soil properties, and the most profitable land allocation should be almost the same considering accessibility, water resources and costs. Therefore, instead of considering land parcels (cells) of size one square kilometer, we aggregate them into larger chunks of land which are called clusters. Each cluster will act as a decision making unit in the optimisation model, and its economic data are obtained by summing over the data related to all active cells in that cluster.

We start by a lower resolution, say R_0, and divide Australia into a number of large parcels of land. Consider the set of clusters $C = \{C_k\}$, where $C_{k_i} \cap C_{k_j} = \emptyset$ for $k_i \neq k_j$, and $\bigcup_k C_k = R$. Each C_k is a nonempty subset of R. For each cluster, all the economic data including profits and costs of all nine activities and all 24 food products, in addition to other related quantities and capacities are calculated based on related data of active cells in that cluster. Note that in the aggregated LP model the number of land-use constraints is less than in the original problem. The reason is that in the original model there is one land-use constraint for each cell. However, in the aggregated model we are dealing with clusters of cells, and there is one land-use constraint for each cluster. The number of other constraints remains the same as the original problem. An aggregated LP is constructed as explained in section 4 with far few-

er variables and constraints, and the optimal primal and dual solutions are obtained by solving the problem using CPLEX.

The plan is to improve the aggregated models consecutively to become as close as possible to the original LP with highest resolution but without solving a large scale problem. Towards this aim, one possible way is to subdivide large clusters and construct smaller ones, and introduce new clusters into the model as new columns. We incorporated reduced costs to separate cells with positive amount inside a cluster from the rest of cells in that cluster. Reduced costs are available at the end of optimisation process. However, these reduced costs correspond to the clusters, and we need to extend this concept to cells inside each cluster.

If, for example, $r_{k,ag}$ is the reduced cost of the variable that corresponds to agricultural activity in the cluster k, we need to distribute it between the agricultural activity of all cells inside the cluster k so that the total sum is equal to $r_{k,ag}$. In other words,

$$\sum_{c \in C_k} r_{k_c,ag} = r_{k,ag}.$$

In this manner, all reduced costs corresponding to all variables related to all activities for each cell in cluster k are calculated. However, before the calculation of reduced costs, we introduce a new concept of semi-reduced costs which is part of the final reduced costs and will be formally defined in the next paragraph. For each cell all semi-reduced costs are calculated for all activities, and the maximum of them is recorded. This quantity represents, to some extent, the most profitable activity for a particular cell. This quantity is also used to distribute the reduced cost of the cluster between all cells proportional to the size of its most profitable activity. The reduced costs are calculated based on semi-reduced costs, and by means of the sign of final reduced costs we can partition each cluster into two sub-clusters containing cells with positive and non-positive reduced costs. The economical quantities for newly constructed clusters are calculated using the information of cells belonging to it, and related information for shrunk clusters are modified considering cells remaining in those clusters. The changes in the old cluster transferred into the LP, and a new column is introduced for each new cluster.

Mathematically, in iteration t, after solving LP_t, we obtain optimal primal solution x_t^* and optimal dual solution π_t^*. At this stage, it is easy to calculate the reduced cost of each activity j of each cluster k using $r_{k_j} = c_{k_j} - \pi_t^* \boldsymbol{p}_{k_j}$. To calculate reduced costs of each cell, perhaps the easiest way is to distribute r_{k_j} uniformly between all cells included in cluster k. However, this approach treats all the cells with equity, and there is not a distinction based on profitability. Another, smarter approach is to distribute r_{k_j} in a way that supports cells with higher profit potentials. In order to calculate reduced cost of each cell in each cluster, first we calculate semi-reduced costs, $\widehat{r_{c_j}}$, for the problem in this paper. The semi-reduced cost is the reduced cost without the portion related to the optimal dual value of the corresponding land-use constraint. In other words, for cell c in cluster k, $r_{c_j} = \widehat{r_{c_j}} - \alpha_c \pi_k^l$. For a cell c in cluster k, if $\max_j\{\widehat{r_{c_j}}\}$ is positive for activity j, the quantity α_c is defined as

$$\alpha_c = \max_j\{0, \frac{\max_j\{\widehat{r_{c_j}}\}}{\sum_c \max_j\{\widehat{r_{c_j}}\}}\}$$

After calculation of reduced costs of each cell in each cluster, the cluster is divided into two sub-clusters. One sub-cluster contains all the cells with positive reduced costs, and the other one contains all the cells with non-positive reduced costs. The economic data of each cluster is updated and new clusters are added into the previous aggregated model as new columns. The model is solved again, and the process of dividing clusters is continued until further bifurcation is impossible.

Algorithm 1 Aggregation and Column Generation

1: Make clusters with a low resolution R_0, and calculate economic attributes of each cluster.
2: Build the aggregated LP, and get optimal primal and dual solutions.
3: Calculate reduced costs of each activity for each cell in each cluster, and subdivide each cluster into two clusters based on the sign of reduced costs of containing cells.
4: If no new clusters created (no cells with negative reduced cost) then the solution is optimal. Stop.
5: Calculate economic attributes of new clusters, and update corresponding data in old clusters.
6: Improve the aggregated LP and include new columns for each recently created cluster, and get the optimal primal and dual solutions.
7: If a stopping criterion is not met, go to Step 3.

6 Computational Results

In this section we present some of the numerical results we obtained through working on application of our algorithm to solve the large scale linear programming problem originated from land use allocation. As seen in Fig. 2 solving a LP problem for each year takes a long amount of time averaging 36 minutes. For a fare comparison, all the numerical experiments are carried out on the same computer. It is worthwhile to mention that the optimisation technique discussed in this paper and all linear programming problems were coded in Python 2.7 language and executed on a computer running a dual core 64-bit Intel(R) Xeon(R) processor at 2.79 GHz with 64 GB RAM. The linear programming problems were solved using IBM CPLEX 12.5 solver.

Table 2 summarizes the output of solving the large scale optimisation problem starting with different resolutions. For our algorithm the stopping criterion is met if we are unable to subdivide the existing clusters further into new, smaller clusters. Resolution 1 is the finest resolution and creates the largest problem. Also, the resolution level of 7569 divides Australia into 417 clusters at the beginning of the computation, and creates the smallest possible LP problem in our experiments.

We recorded starting number of clusters, n_{start}, final number of clusters, n_{end}, the number of iterations to obtain the final optimal value, and the percentage difference between our optimal value with the optimal value of the original problem for each resolution. The original problem in its finest resolution of 1 has 812,383 clusters, due to considering each cell as a cluster. The algorithm could find the optimal solution in one iteration, as it is impossible to break down clusters. We also record the amount of time taken for CPLEX to find the optimal solution for each resolution in minutes. This column provides us a good measure to compare time performance of our approach. We did not consider the time necessary for the extraction and calculation of parameters in this column, and in the execution time provided in section 5.

resolution	iterations	n_{start}	n_{end}	Difference (%)	t (min)
1	1	812383	812383	0.0	53.598
9	4	138329	138368	0.16	25.930
81	5	20402	20482	0.27	0.694
784	5	2784	2899	0.27	0.020
1600	7	1608	1742	0.26	0.016
7569	7	417	484	0.49	0.009

Table 2. Results on running the algorithm of different resolutions

The data in the table demonstrates some interesting facts about our approach and its capability to deal with large scale linear programming problems. Instead of solving the largest possible problem, the algorithm tries to solve an aggregated problem which is a considerably smaller problem. After each iteration, it tries to subdivide clusters based on their reduced costs. As an example, starting with resolution 7,569 which divides Australia into 417 clusters at the beginning, only in seven iterations and with addition of 67 clusters, the algorithm could reach to the optimal objective function with only 0.49 percent deviation in a small portion of a minute. Comparing this outcome with solving the original problem which needs 53 minutes shows how fast the new algorithm could find the optimal solution. Furthermore, Fig. 3 represents the convergence rate to the optimal solution considering each resolution factor.

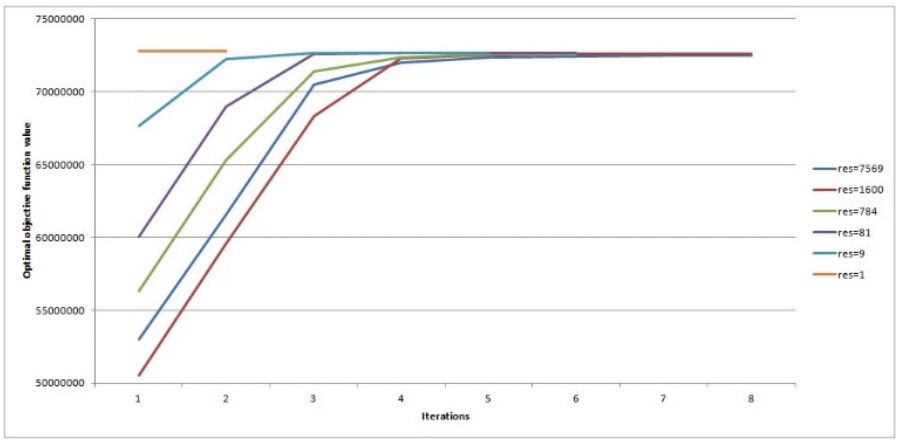

Fig. 3. Convergence results for each resolution

It is interesting to notice that choosing a lower resolution of 1,600 causes to start with a worse bound than 7,569. However, as shown in the Fig 3 it gets better quick, meaning that less clusters do not necessarily mean worse bounds, and also the smaller clusters are not necessarily all subsets of the larger clusters.

Combining the information presented in Table 2 and Fig 3 it is apparent that there is a smarter approach of solving the large scale LP in a noticeably shorter amount of time by choosing a lower resolution at the beginning and gradually refining the resolution of clusters in the sequel iterations. With this approach, it is meaningful to build a stochastic optimisation model of the land allocation planning problem, and obtain sensible outcomes in reasonable amount of time and effort.

7 Conclusion

In this paper we presented a new efficient algorithm to solve large scale linear programming problem originated from optimal land allocation planning. The algorithm combines techniques of column generation, aggregation and disaggregation particularly suitable for this problem. We approximated the large problem with a small LP problem by considering huge parcels of land as a cluster, and then tried to improve the approximation iteratively by subdividing clusters when it is possible. The use of our custom aggregation and disaggregation method allows us to easily solve land use models at a much higher resolution than would otherwise be possible, thus improving the fidelity of the models. In addition, the huge reduction in computational time lays the ground for more complex models. For example an extension to stochastic optimisation incorporating multiple scenarios in a single model is being considered.

References

1. Dantzig G.,Wolfe P.: Decomposition principle for linear programs. Oper. Res. 8(1), 1960, pp. 101–111.
2. Gilmore P.C., Gomory R.E.: A linear programming approach to the cutting-stock problem. Oper. Res. 9(6), 1961, pp. 849–859.
3. Desaulniers G., Desrosiers J., Solomon M.: Column Generation, Springer, New York, 2005.
4. Lübbecke M.E., Desrosiers J.: Selected topics in column generation. Oper. Res. 53(6), 2005, pp. 1007–1023.
5. Litvinchev I., Tsurkov V.: Aggregation in Large-Scale Optimization, 2003, Springer.
6. Rogers D.F., Plante R.D., Wong R T., Evans J.R.: Aggregation and Disaggregation Techniques and Methodology in Optimization, Operations Research Vol. 39, No. 4,1991, pp. 553–582.
7. Weintraub A., Cholaky A.: A Hierarchical Approach to Forest Planning, Forest Science, Volume 37, Number 2, 1991, pp. 439–460.
8. Seppelt R., Lautenbach S., Volk M.: Identifying trade-offs between ecosystem services, land use, and biodiversity: a plea for combining scenario analysis and optimization on different spatial scales, Current Opinion in Environmental Sustainability, 5(5), 2013, pp. 458–463.
9. Gaddis E.J.B., Voinov A., Seppelt R., and Rizzo D.M.: Spatial Optimization of Best Management Practices to Attain Water Quality Targets, Water Resources Management, 28 (6), 2014, pp. 1485–1499.
10. Klein T., Holzkämper A., Calanca P., Seppelt R., Fuhrer J.: Adapting agricultural land management to climate change: a regional multi-objective optimization approach, Landscape Ecology, 28(10), 2013, pp. 2029–2047.
11. Bryan B.A., Nolan M., Harwood T.D., Connor J.D., Navarro-Garcia J., King D., Summers D.M., Newth D., Cai Y., Grigg N., Harman I., Crossman N.D., Grundy M.J., Finnigan J.J., Ferrier S., Williams K.J., Wilson K.A., Law E.A., Hatfield-Dodds S.: Supply of carbon sequestration and biodiversity services from Australia's agricultural land under global change, Global Environmental Change, 28, 2014, pp. 166–181.