

WHAT-IF ANALYSIS THROUGH SIMULATION-OPTIMIZATION HYBRIDS

Marco Gavanelli
ENDIF University of Ferrara, Italy
marco.gavanelli@unife.it

Michela Milano
DEIS University of Bologna, Italy
michela.milano@unibo.it

Alan Holland and Barry O'Sullivan
Cork Constraint Computation Center, Ireland
{a.holland,b.osullivan}@4c.ucc.ie

ABSTRACT

This paper proposes to improve traditional what-if analysis for policy making by a novel integration of different components. When a simulator is available, a human expert, e.g., a policy maker, might understand the impact of her choices by running a simulator on a set of scenarios of interest. In many cases, when the number of scenarios is exponential in the number of choices, identifying the scenarios of interest might be particularly challenging. We claim that abandoning this *generate and test* approach could greatly enhance the decision process and the quality of political actions undertaken. In this paper we propose and experiment with one approach for combining simulation with a combinatorial optimization and decision making component. In addition, we propose two alternative approaches that can reasonably combine decision making with simulation in a coherent way and avoid the generate and test behaviour.

KEYWORDS

POLICY MODELING SOCIAL SIMULATION COMBINATORIAL OPTIMIZATION

INTRODUCTION

Public policy issues are extremely complex, occur in rapidly changing environments characterized by uncertainty, and involve conflicts among different interests. Our society is ever more complex due to globalisation, enlargement and rapidly changing geo-political situation. This means that political activity and intervention become more widespread. Therefore, the effects of any such interventions become more difficult to assess. Of course, it is becoming ever more important to ensure that actions are effectively tackling the real challenges that this increasing complexity entails. Thus, those responsible for creating, implementing, and enforcing policies must be able to reach decisions about ill-defined problem situations that are not well understood, have no one correct answer, involve many competing interests, and interact with other policies at multiple levels. It is therefore more important to ensure coherence across these complex issues.

The majority of policy models rely on agent-based simulation (Troitzsch et al., 1999; Matthews et al., 2007;

Gilbert, 2010) where agents represent the parties involved in the decision-making and implementation process. The hypothesis is that for modelling complex systems, agent-based simulation is a suitable approach to understand such systems in a more natural way. In particular, agent-based models enable the use of computer experiments to support a better understanding of the complexity of economic, environmental and social systems, structural changes, and endogenous adjustment reactions in response to a policy change. In addition to agent-based simulation models, which provide *individual level models*, we claim that the policy planning activity needs a global perspective that faces the problem at a global level and should tightly interact with the individual level model. The policy maker must take decisions by perceiving a set of (possibly conflicting) objectives, and satisfying a set of constraints while at the same time reducing negative impacts and enhancing positive impacts on the environment, society and economy. Simulation could be therefore used to understand the impact of her decisions via what-if analysis or scenario analysis.

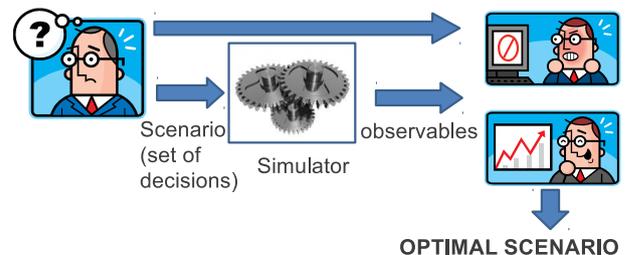


Figure 1: Manual decision making

If a decision maker received no useful feedback regarding the impact of decisions, this would be the worst outcome of all. However, typically at present a policy maker devises a set of scenarios to be simulated and evaluates the impact of the taken decision. The process iterates as soon as the policy maker finds a solution that satisfies her, as depicted in Figure 1. The simplest way to improve this process is to aid the decision maker in the first step of her process, by designing a Decision Support System (DSS) for the selection of (Pareto) optimal points corresponding to specific political actions, as depicted in Figure 2. In brief, the problem components that the policy maker should take into account, namely impacts on

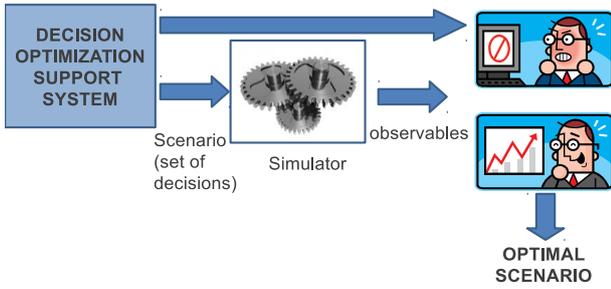


Figure 2: Decision support system for scenario selection

environment, economy, financial aspects, territory-based constraints, and objectives, can be cast as a combinatorial optimization and decision problem and solved using the appropriate techniques, described e.g. in (Gavanelli et al., 2010, 2011). Despite being more sophisticated than the *manual* approach, this secondary process still exhibits a generate-and-test behaviour. In other words, the decision making and optimization component operates within the confines of a limited information set and is not guided toward simulation-reasonable solutions.

In this paper we propose an approach that enables a tight interaction between a simulator and a decision making component based on machine learning. Machine learning is used to synthesize constraints for the decision making component from simulation results. We show an example applied to the Italian Emilia-Romagna region, and in particular on its Regional Energy Plan. The definition of the plans is provided by a decision support system by casting the problem in a mathematical model and solving it through optimization techniques. On the other hand, the implementation strategy, namely the definition of the percentage of incentives to reach the objectives of the regional plan, can be only understood through agent-based simulation. The proper interaction between policy planning and policy implementation should necessarily lead to an integration of decision support and simulation components.

Also, we devise two alternative approaches that maintain close connections between simulation and the decision-making components, and abandon the generate and test behaviour: one is based on Benders' decomposition, while the other is based on the game theoretic principles of Mechanism Design.

The paper is organized as follows: we first describe the considered case study, i.e. the regional energy plan, then we propose the first approach of optimization-simulation hybrids we experimented with, namely the one using machine learning techniques. Then we propose two alternative approaches that would reasonably improve the first integration scheme: one based on problem decomposition and one on mechanism design.

THE REGIONAL ENERGY PLAN CASE STUDY

The Regional Energy Plan defines the strategic regional objectives for the energy production and energy effi-

ciency. The case study is based on data provided by the Italian Emilia-Romagna region. The region has strategic objectives, financial and territorial constraints and environmental impacts. An example of an objective that the region might have is to increase the production of energy while at the same time increasing the share of renewable energy sources in the regional energy balance.

For each energy source, the plan should provide: the installed power, in MW; the total energy produced in a year, in kTOE (TOE stands for Tonne of Oil Equivalent); the total cost, in M€. The ratio between installed power and total produced energy is mainly influenced by the availability of the source: while a biomass plant can, at least in theory, produce energy 24/7, the sun is available only during the day, and the wind only occasionally. For unreliable sources an average for the whole year is taken. The cost of the plant, instead, depends mainly on the installed power: a solar plant has an installation cost that depends on the surface area of installed panels, which on their turn can provide some maximum power (peak power).

The 20-20-20 EU directive imposes that 20% of total energy requirements for 2020 should be provided by renewable sources. Technicians in the region proposed a percentage to be provided during the period 2011-2013 of 177kTOE of electrical energy and 296kTOE of thermal energy. With this premise they developed plans for electrical and thermal energy, respectively. In Table 1 we show the electric plan based only on renewable energy sources. The total private and public investment requirement is 3014M€.

Table 1: Regional Energy plan 2011-2013

	Power 2010 (MW)	Power 2013 (MW)	Energy 2013 (kTOE)
Power plants			
Hydroelectric	300	310	69.3
Photovoltaic	230	850	87.7
Th.dyn. solar	0	10	1
Wind farms	20	80	10.3
Biomasses	430	600	361.2
Total	980	1850	529.5

In general, after a plan is created, the policy maker defines actions for its implementation. A widely used instrument for supporting the renewable energy market is monetary incentives. For example, in the Emilia-Romagna region, incentives for the photovoltaic energy are distributed to stakeholders by means of auctions that indeed do not result from a specific strategy, but rather from extemporaneous strategies. In these auctions the bids are ranked on the basis of various criteria (including the co-financing percentage), and the first n bids that satisfy the budget constraint are funded. However, this strategy may be far from optimal and further research is required to examine the efficacy of this approach.

Italian law considers four different methods for pro-

viding incentives, while in the auctions developed in the past only one type was used. Finally, one should consider carefully how to spend the limited financial resources available: for example, should we provide higher incentives or spend more money into advertising the availability of incentives? As we can see, the number of possible choices, just for the implementation of the plan, is combinatorial, which makes a hand-made process inapplicable or highly sub-optimal. Moreover, the outcome of a given policy of incentives is not clear or easily foreseeable, and it can only be obtained through a social simulator.

In this paper, we propose to integrate in the planning process a mechanism for defining a proper incentive strategy that achieves the objectives of the plan. For this reason we should devise a methodology for coupling a decision component with a simulator.

INTEGRATING SIMULATION WITH LEARNING

We report on some experiments we have conducted that show how a machine learning system can be integrated with a simulator to assist in the policy-making process associated with a regional energy planning task. In this case, the simulator generates a set of scenarios, relating decisions with observables as depicted in Figure 3. The collected tuples $\langle decision_1, \dots, decision_n, observable_1, \dots, observable_m \rangle$ are stored as a training set for a learning component which in turn learns a relation between decisions and observables. The relation can be an objective function or a constraint or a cost function, the automated modeling of which can be framed as a regression problem. In any case it should have a form that is compatible with the decision support system model.

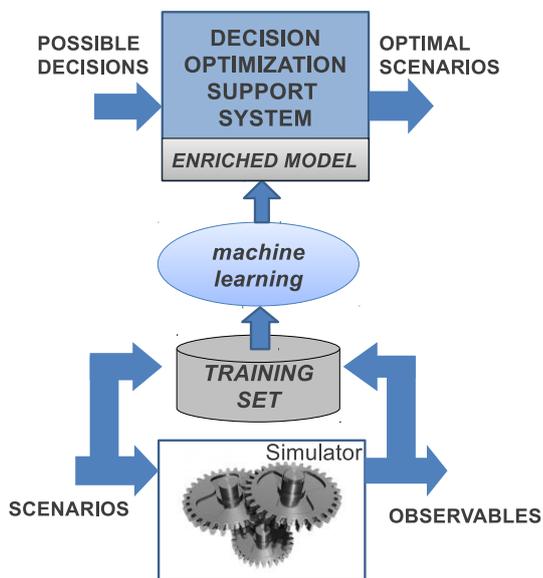


Figure 3: Learning-Based Interaction

We consider now the learning based interaction. What we want to learn is a function that relates the installed power to the percentage of incentives provided by the region. We have performed a large number of simulations

(1500) for each value of incentives from 1% to 30% in steps of 1% (for a total of 45,000 simulations). Each simulation has 1000 agents, and we recorded, for each simulation, the total installed power in MW of photo-voltaic plants. A plot of the results is shown in Figure 4 where each point represents an individual simulation.

Of course, an individual simulation does not provide useful information, as can be seen from Figure 4, and one should try to extract some statistics from a significant number of simulations in order to get some insight. In order to learn a model of the dependency of the installed power from the incentives, we average the results of all the simulations with the same amount of incentives. In this way we obtain a point for each value of the incentives from 1% to 30%. Then we learned a function that relates the installed power to the incentives. We tried various regression algorithms, including linear regression (Rousseeuw and Leroy, 1987), Gaussian processes (Mackay, 1998), least median squared linear regression (Rousseeuw and Leroy, 1987), multilayer perceptron (Mitchell, 1997), Gaussian radial basis function networks (Mitchell, 1997) and support vector regression (Smola and Schölkopf, 2004).

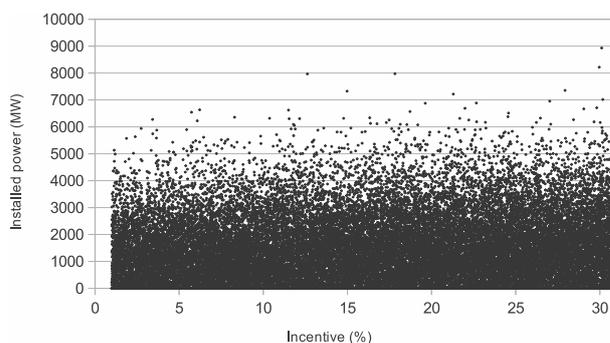


Figure 4: Simulations

We evaluated the mean squared error of each algorithm using ten-fold cross validation. The algorithm that gave the lowest mean squared error was linear regression so we applied it to the whole dataset and we obtained the function

$$G_{PV} = mI\% + q \quad (1)$$

(where PV stands for photovoltaic) that is also shown in Figure 5. The values of the parameters we obtained are $m = 2645MW$ and $q = 405MW$.

We inserted in the model the relationship of Equation 1, in this way we were able to relate the incentives given by the region with the obtained installed power of photovoltaic energy.

The region has a limited budget B_{PV} for the incentives towards photovoltaic plants (PV), so this should be imposed as an upper bound to the total incentive:

$$I^{Tot}_{PV} \leq B_{PV} \quad (2)$$

The total incentive is given by the percentage of incentives given by the region multiplied by the total cost of

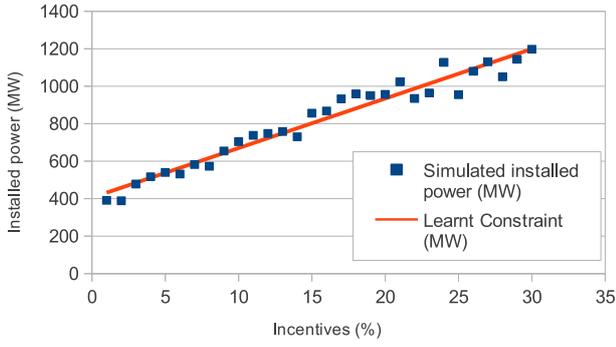


Figure 5: Learned function.

installed photovoltaic panels:

$$I^{Tot}_{PV} = I\% c_{PV}^{Tot} \quad (3)$$

the cost is given by the unit cost c_{PV} multiplied by the magnitude of the installed panels:

$$c_{PV}^{Tot} = c_{PV} G_{PV} \quad (4)$$

From the simulations and linear regression, we obtained the relationship between the provided percentage of incentives and the expected magnitude (Eq. 1). Combining equations (1), (3) and (4) we obtain:

$$I^{Tot}_{PV} = I\% c_{PV} (mI\% + q) = c_{PV} m (I\%)^2 + qI\% \quad (5)$$

The constraint of Eq. (2) can be rewritten (solving Eq. (5) for $I\%$ and excluding a trivial bound, since $I\%$ is non-negative), as:

$$I\% \leq \frac{-q + \sqrt{q^2 + 4mB_{PV}/c_{PV}}}{2m}. \quad (6)$$

By inserting this constraint into the decision support system, we can define regional energy plans whose incentives strategies are compatible with budget constraints.

MORE SOPHISTICATED INTEGRATIONS

In this section we consider how more sophisticated hybrids of optimisation and simulation can be developed. In the first case we consider an approach based on a classic problem decomposition technique from the field of operations research. This technique can be used to link the multiple levels of abstraction needed in a complex policy-making setting. In the second scenario we consider how game theoretic concerns can be considered in the process, by exploiting the utilities of agents to design complex incentive schemes to motivate particular behaviours that maximise overall efficiency of the system.

Benders Decomposition

Benders (1962) decomposition is a method for solving combinatorial optimization problems that can be decomposed into two components: a master problem and a subproblem. Benders Decomposition was originally devised

in the field of Integer Linear Programming, but has been extended for dealing with general solvers in the so called Logic-Based Benders Decomposition (Hooker and Ottosson, 2003). In our case, the master problem is the definition of the regional energy plan that partitions the needed energy into renewable sources. The master problem is solved through Constraint Programming as described in (Gavanelli et al., 2012). The subproblem is the definition of the incentive strategy to achieve the required photovoltaic installed power, that is also consistent with the regional budget. The simulator is run to understand which is the proper required incentive to obtain the solution provided by the master problem. In case the incentives are not compatible with the regional budget a Benders cut is generated and another solution is provided by the master.

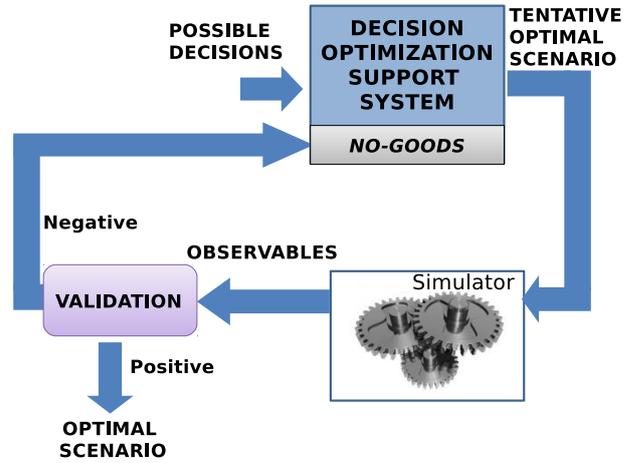


Figure 6: Benders Decomposition Interaction

The learning-based interaction, as described in previous section, requires the execution of a very high number of simulations in order to (1) get significant statistics for each value of the incentives (or, in general, for each possible decision in the policy) and (2) provide a wide set of data for the machine learning.

On the other hand, it may be the case that some values of policy decision variables are not interesting, as they would provide very bad values for the Decision Support System that utilizes optimization. In principle, one would like to simulate only the best values from the DSS viewpoint; unluckily these values are unknown and depend on the constraints provided by the machine learning component. This shows that the architecture in Figure 3, with one-way communication from the simulator to the DSS, should be extended to a cycle, that provides bidirectional communication between the two main components (Figure 6).

The interaction starts from the DSS, that generates an optimal solution of the master problem. This solution contains tentative values for the incentives, and for the required outcome for photovoltaic energy. The tentative values are passed to the simulator, that executes a number of simulations only for those values of the param-

ters provided by the DSS, and provides the correspondent statistics. These statistics might confirm or not the tentative values proposed by the DSS: if the (average) simulated outcome is higher or equal to the corresponding tentative value, the iteration stops and the result is provably optimal (Benders, 1962). Instead, if the tentative value of the outcome is higher than the simulated value, another iteration is required. So, a constraint (or *nogood*) is communicated from the simulator to the DSS, explaining that one cannot obtain the requested level of photovoltaic power with the proposed value of incentives. The DSS inserts the *nogood* into the constraint model, solves it to optimality, and provides new tentative values to the simulator.

The main challenge is determining the set of constraints that are communicated between the two systems: if one excludes from the feasible set only the tentative values, the risk is to perform many interactions, leading to exhaustive simulation of all the values of the parameters, while if one excludes further values there is the risk to possibly discard promising solutions. This issue is subject of current research.

INCENTIVE COMPATIBLE MECHANISMS

We now consider an approach that supports What-If Analysis based upon the principles of Economic Theory and, in particular, Game Theory. When one assumes that agents are all self-interested and rational utility maximizers, one can apply the solution concepts of Nash Equilibrium to predict expected outcomes. This is an attractive concept for policy makers because it can aid the predictability of novel economic policies or initiatives.

A seminal result known as the *Revelation Principle* states that, no matter the mechanism, a designer concerned with efficiency need only consider equilibria in which agents truthfully report their “types” that signify their private valuation for an item (Gibbons, 1992). Cleverly designed economic mechanisms (or auctions) can allocate resources and determine payments that are resilient to manipulation. The design of subsidy schemes to support the construction of public goods is particularly challenging (Laffont, 1987). Our setting involves a possibly large number of agents and a set of renewable technologies so tractability concerns must also be borne in mind (Nisan and Ronen, 2001). The key design challenge concerns the *free rider problem* and consequent under provision of public goods. For example, in first price auctions previously conducted in the Emilia-Romagna region, participants that wished to acquire a photovoltaic device without government aid had an incentive to under-report their valuation to receive a subsidy.

Mechanism design is a game of private information in which a single central agent, the “center”, chooses the payoff structure. Agents report a type to the center that they may choose strategically so that it is different from their true value. After the reporting phase, the center determines an outcome. The outcome consists of an allo-

cation and a payoff. The center typically wishes to fulfil a *social choice function* to map the true type profile directly to the allocation of goods transferred, whereas a *mechanism* maps the reported type profile to an outcome.

A Subsidy Disbursement Example

Let us consider a more specific example. There are m households whose suitability for receiving a subsidy for a photovoltaic device depends upon the pitch of their roof, orientation, horizon profile and location. We reduce these parameters to a single value describing the worth of a solar device per unit of power to value v^i for agent i . This value reflects the day zero time-discounted value of the expected stream of future cashflows given their circumstances. The *social choice function* is to assign the panels p_j , $j \in \{1, \dots, J\}$ to agents in a manner that minimizes the maximum cost for any agent. The imposition (or cost) for agent i if panel j is received is the price of the device minus the value per unit of power multiplied by the power output of the device,

$$c_j^i = r_j - v^i \phi_j, \forall i \in \{1, \dots, I\}, j \in \{1, \dots, J\}$$

where r_j is the purchase price of an installed panel.

This problem can be transposed to a makespan minimization problem denoted in the scheduling theory literature as $Q||C_{max}$, where C_{max} refers to makespan in scheduling theory. We wish to allocate device acquisition and hosting responsibility (jobs) across houses (machines) that each perceive a private cost associated with acceptance of that job. The minimization of the maximum time to wait for all jobs to complete is comparable to the minimization of the maximum cost imposed on any house-owner so that inconvenience is bounded as tightly as possible.

Non-monotone Algorithm Consider the following simple algorithm: order the panels from highest to lowest power and greedily assign each device in turn to the household that has received lowest cost imposition thus far in the partial allocation. This algorithm is a 2-approximation that is non-monotone. Consider an example with 3 devices $\{d_1, d_2, d_3\}$ and 2 agents $\{h_1, h_2\}$ that illustrates non-monotonicity. Let the publicly known power ratings for the devices be $\phi_1 = 10W$ and $\phi_2 = \phi_3 = (9 + \epsilon)W$ and all devices cost $r_j = 60\text{€}$. This is common to all agents. However, let each house-owner’s value per unit of power be $v_1 = 5\text{€/}W$ and $v_2 = (5 - \epsilon)\text{€/}W$. This is the private information that we wish to elicit. Our greedy algorithm first assigns: $d_1 \rightarrow h_1$, $d_2 \rightarrow h_2$ and $d_3 \rightarrow h_2$ resulting in costs $c_1 = 60 - 10 \times 5 = 10\text{€}$ and $c_2 = 2(60 - (45 - 4\epsilon - \epsilon^2)) \cong (30 + 8\epsilon)\text{€}$. But if we increase v_2 so that $v_2 = (5 + \epsilon)\text{€/}W$ then it receives only the first device. The first (highest power device) will be assigned to the second agent because she now has a higher value per unit of power. The second (lower power) device is assigned to the first agent because this agent has received a lower

cost imposition thus far. The third (lower power) device is also assigned to the first agent because this agent has received a lower cost imposition thus far because the previous lower power device imposed less cost than the high power device did for the agent that values renewable power more. So this algorithm is not monotone and a direct consequence of this is that it cannot be used within any truthful mechanism for allocating devices to agents (Archer and Tardos, 2001).

Monotone Algorithm There exists a *randomized* 3-approximation that is truthful in expectation. Kovács (2005) developed an approximation scheme for scheduling n jobs to m machines of different speeds so that the makespan is minimized. This problem is sometimes referred to as $(Q||C_{max})$ (Kovács, 2005). A fast, deterministic *monotone* 3-approximation algorithm exists for this problem. The importance of monotonicity is very relevant to our setting and the context of truthful mechanisms in general. When each agent knows its own value for hosting a device, it is necessary to design an incentive for declaration of true values to enable efficient allocation. Archer and Tardos (2001) demonstrated that such motivation is possible only if the allocation algorithm within the mechanism is monotone.

RELATED WORK

In the literature, simulation and optimization have been merged mainly in the so-called simulation optimization field. In this case optimization aids simulation for choosing optimal parameters to improve operations (Deng, 2007). The goal of optimization routines is to seek improved settings of system parameters with respect to the performance metrics. Similarly, Neuro-Dynamic Programming (NDP) (Bertsekas and Tsitsiklis, 1996) is an approach to select agent decision making rules (feedback policy) that optimize a certain performance criterion. NDP often relies on simulation, in the so called value function approximation, to tune the parameters of a value function that quantifies the relative desirability of different states in the problem space. Markov decision processes have also been used in Reinforcement Learning (RL) together with simulation, for learning directly the policy parameters (Marbach and Tsitsiklis, 2001). Both NDP and RL are concerned with how a single agent ought to take actions in an environment so as to maximize some form of cumulative reward. On the contrary, in this paper we are interested in understanding how global political actions and interventions impact on a complex systems (namely, the energy market) without changing the agent behavior. We aim to (1) observe/learn the causal link between agents behavior and high level decisions and (2) cast these relations into a model component.

Simulation-aware optimization has been considered in the context of Genetic Algorithms (GAs). The basic integration technique consists in solving a numerical model to evaluate the fitness function (see for example

(Obayashi et al., 2000)). Although GAs encode some knowledge of the system behavior through the population individuals, these approaches learn no explicit relation between decision variables and the system observables. As a consequence, analytic properties of the controlled system cannot be discovered and exploited with the typical means of combinatorial optimization.

Finally, the closest approaches to this paper are those related to the governance of a simulated system. The traditional way to cope with combinatorial (global) decision making where a part of the model can be simulated is rather trivial: the decisions are taken and the simulator evaluates their “quality”. This is the base of Simulation for Optimization (Fu, 1994) that is strongly based on stochastic programming. The main idea is that candidate solutions are presented to the stochastic discrete-event simulator that, in turn, provides performance estimates of the solutions via statistical analysis. In this case the simulator model is another objective function generator, but the way solutions are presented follows a pure generate-and-test pattern. A similar approach is considered in OptQuest (Glover et al., 1999), a system that integrates in a closed loop simulation and metaheuristics to achieve good quality solutions. In the paper also a primitive form of learning is used, namely a neural network accelerator aimed at avoiding trivially bad solutions. In the same fashion, in (Bartolini et al., 2011), simulation has been used to learn neuron constraints in a thermal aware dispatching application. This last paper is close to the one presented here, with the difference that the learning component aims at tuning parameters of neural constraints, while in this paper we learn a linear function linking decision variables and observables that can be evaluated and validated by domain experts.

CONCLUSION AND FUTURE WORK

In future work we aim to tackle the challenge of devising a mechanism that will facilitate a tractable and budget balanced approach to allocation and payments. We plan to consider alternative social choice functions that takes a more holistic view of agent types that considers both welfare and aggregate renewable energy generated.

We face in this paper the very challenging problem of mixing a regional planning activity with the definition of an implementation strategy. While the planning part can be cast into a combinatorial optimization problem and solved through optimization techniques, the implementation strategy requires a simulator to be understood involving self interested decision making agents. We propose here one technique for integrating optimization and simulation and two alternative solutions to aid this process that are subject of current study in the context of the EU FP7 *ePolicy* Project.

ACKNOWLEDGEMENTS

We would like to thank Fabrizio Riguzzi for his help on the machine learning part.

This work was partially supported by EU project *ePolicy*, FP7-ICT-2011-7, grant agreement 288147. Possible inaccuracies of information are under the responsibility of the project team. The text reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained in this paper.

REFERENCES

- Archer, A. and Tardos, Éva. (2001). Truthful mechanisms for one-parameter agents. In *Proceedings of FOCS*.
- Bartolini, A., Lombardi, M., Milano, M., and Benini, L. (2011). Neuron constraints to model complex real-world problems. In *Principles and Practice of Constraint Programming*.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252.
- Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.
- Deng, G. (2007). *Simulation-based Optimization*. PhD thesis, University of Wisconsin - Madison.
- Fu, M. (1994). Optimization via simulation: A review. *Annals of Operations Research*, 53:199–248.
- Gavanelli, M., Riguzzi, F., Milano, M., and Cagnoli, P. (2010). Logic-Based Decision Support for Strategic Environmental Assessment. *Theory and Practice of Logic Programming*, 10(4-6):643–658.
- Gavanelli, M., Riguzzi, F., Milano, M., and Cagnoli, P. (2012). Constraint and optimization techniques for supporting policy making. In *Computational Intelligent Data Analysis for Sustainable Development*, chapter 16. Taylor & Francis.
- Gavanelli, M., Riguzzi, F., Milano, M., Sottara, D., Cangini, A., and Cagnoli, P. (2011). An application of fuzzy logic to strategic environmental assessment. In Pirrone, R. and Sorbello, F., editors, *AI*IA*, volume 6934 of *LNCS*. Springer.
- Gibbons, R. (1992). *Game theory for applied economists*. Princeton University Press.
- Gilbert, N. (2010). *Computational Social Science*. SAGE.
- Glover, F., Kelly, J., and Laguna, M. (1999). New advances for wedding optimization and simulation. In *Proc. of the Winter Simulation Conference*.
- Hooker, J. N. and Ottosson, G. (2003). Logic-based benders decomposition. *Mathematical Programming*, 96:33–60.
- Kovács, A. (2005). Fast monotone 3-approximation algorithm for scheduling related machines. In *Proceedings of the 13th annual European conference on Algorithms*, ESA'05, pages 616–627, Berlin, Heidelberg. Springer-Verlag.
- Laffont, J.-J. (1987). Incentives and the allocation of public goods. *Handbook of public economics*, 2:537–569.
- Mackay, D. J. (1998). Introduction to Gaussian processes.
- Marbach, P. and Tsitsiklis, J. N. (2001). Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209.
- Matthews, R., Gilbert, N., Roach, A., Polhill, G., and Gotts, N. (2007). Agent-based land-use models: a review of applications. *Landscape Ecology*, 22(10).
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.
- Nisan, N. and Ronen, A. (2001). Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196.
- Obayashi, S., Sasaki, D., Takeguchi, Y., and Hirose, N. (2000). Multiobjective evolutionary computation for supersonic wing-shape optimization. *IEEE Transactions on Evolutionary Computation*, 4:182–187.
- Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection*. Wiley.
- Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14:199–222.
- Troitzsch, K. G., Mueller, U., Gilbert, G. N., and Doran, J. (1999). Social science microsimulation. *J. Artificial Societies and Social Simulation*, 2(1).

AUTHOR BIOGRAPHIES

MARCO GAVANELLI is Ricercatore (Assistant Professor) in Computer Science at the Department of Engineering, University of Ferrara, Italy. His research interests are on Logic Programming and Constraint Programming and their applications. His personal webpage is at <http://www.ing.unife.it/docenti/MarcoGavanelli/>.

BARRY O’SULLIVAN holds the Chair in Constraint Programming at University College Cork and is Director of the Cork Constraint Computation Centre (4C) in the Department of Computer Science. His personal webpage at <http://osullivan.ucc.ie>.

ALAN HOLLAND is a Research Fellow specialising in optimisation and electronic commerce in the Cork Constraint Computation Centre (4C), Department of Computer Science, University College Cork, Ireland. His personal webpage at <http://4c.ucc.ie/~aholland>.

MICHELA MILANO is Associate Professor in Intelligent Systems at the Department of Electronic, Computer Science and Systems, University of Bologna, Italy. Her research interests span from Artificial Intelligence to Operations Research to build hybrid optimization techniques. Her personal webpage at <http://ai.unibo.it/people/MichelaMilano>.