Decision Making on Complex Systems: Challenges and Opportunities

Michela Milano and Michele Lombardi

DISI, University of Bologna
V.le Risorgimento 2, 40136, Bologna, Italy

Abstract. In this paper, we propose a challenging research direction for Constraint Programming and optimization techniques in general. We address problems where decisions to be taken affect and are affected by complex systems, that exhibit phenomena emerging from a collection of interacting objects, capable to self organize and to adapt their behaviour according to their history and feedback. Such systems are unfortunately impervious to modeling efforts via state-of-the-art combinatorial optimization techniques.
We provide some hints on methodologies to connect and integrate decision making and optimization technology with complex systems via machine learning, aimed at extracting modeling components to express the relation between global decisions and observables emerging from simulation, and game theory, to drive the process toward realistic equilibrium points instead of unrealistic, globally optimal, configurations. The capabilities of state of the art solvers should be greatly extended and enhanced to cope with the complexity of the considered problems.

Introduction

Combinatorial decision making and optimization technology has achieved a good level of maturity in the last decades. A number of different approaches have been defined to effectively and efficiently solve decision and optimisation problems: exact methods relying on Mathematical Programming [27], constraint-based reasoning and search [24] and incomplete methods ranging from simple local search to more sophisticated meta-heuristics [8].

Recently, these techniques have been properly merged and hybridized to face real life structured problems [25], [26]. Each technique faces different problem aspects, criteria or components.

A large number of problems from industry, business, manufacturing, and science are now within the reach of combinatorial optimization techniques. Examples of application areas are scheduling, timetabling, biology, system design, configuration and resource allocation.

Despite significant advances in algorithmic research, however, the current state of decision support and optimization solvers still faces severe difficulties or cannot cope at all with problems that arise from decision making on complex systems. Complex systems have been widely studied in various disciplines such as physics [2], biology [20], economy [22], but they all share some common characteristics such as rapid and unforeseen transitions between order and disorder, interactions between a number of decision making (possibly cognitive) agents, evolution of the overall behaviour on the basis of feedback from previous experience. The most traditional way to study these systems is through mathematical modeling and simulation whose aim is to observe, predict and even control such systems [32]. When we have to take decisions in presence of a complex system, simulation can be used to aid the decision maker through the so called what-if analysis. The process follows a generate and test pattern: the decision maker selects, on the basis of her experience, a very limited subset of decision scenarios, that are possibly very far from the optimal ones, runs the simulator on these scenarios and observes the result through the so-called observables. In case the decision problem is combinatorial in nature, the process could be very long and error prone.

To better clarify the above concept, consider a policy maker who has to define a regional energy plan for her region. Beside a planning strategy and resource allocation for renewable energy sources, that can be modeled as a combinatorial optimization problem, she has to define proper incentives to foster the use of renewable energy sources. Given the regional objectives as constraints, she wants to obtain a given increase in the use of renewable energy, but plants are built by citizens and enterprises. Therefore she has to understand how much money and how this money is distributed to push citizens and enterprises to install renewable energy plants to obtain the desired increase. So the policy maker has to consider in her decision process the energy market that is a complex system, understand its trend and take decisions in order to drive the market toward the desired direction. One might think that the problem is easy: the higher the incentives, the higher the energy installed. This would be true if only economical aspects are considered by agents. The real market behaviour is indeed different. Agents are driven also by social and cognitive aspects such as belongingness to a group, freedom from the power suppliers, trust in future and government, perceived bureaucracy and these aspects depend also on the number of contacts of each agent. These aspects are difficult to predict and insert into a combinatorial problem model.

In these settings, decisions to be taken affect and are affected by complex systems [33]. These systems exhibit phenomena which emerge from a collection of interacting objects that are able to self-organize and are affected by memory or feedback so that they can adapt their behaviour according to their history. Complex systems exhibit emergent behaviours which are generally surprising, and may be extreme. These emergent behaviours typically arise in the absence of any sort of invisible hand or central controller. In addition, the systems show a complicated mix of ordered and disordered behaviour. Therefore what can be measured in a complex system is a kind of "average behaviour" or a "trend" trough a set of figures called observables that can be extracted either from the system itself, or from a simulator.

Current decision making techniques cannot cope at all with complex systems because we need an accurate representation for the target domain. As a matter of fact, many optimization approaches assume the availability of a declarative description of the system, usually obtained through the interaction between a modeling expert and a domain expert by introducing some degree of approximation; the resulting accuracy is critical for the optimization effectiveness: an over-simplified model may threat the successful application of the most advanced combinatorial methods, while accurate models could be computationally intractable. In addition, coming up with an accurate model may be very challenging whenever (1) there are elements admitting no obvious numerical description, or (2) the system behaviour results from the interaction of a very large number of actors as happens in complex systems.

To enable Constraint Programming and more in general hybrid methods to cope with these systems we need a broad multi-disciplinary approach that can bring complex systems related problems within the reach of combinatorial decision making and optimization. What we have to include into the combinatorial model are relations between the observables of the complex systems and the decisions taken. We have to extract (part of) a decision making and optimization model from the system itself, or better, from the simulation. The model should describe the relations between the decision variables and their expected effect on the complex system it interacts with.

To obtain such integration, we need to enlarge the scope of optimization techniques to make use of machine learning, and game theory. In this paper, we provide some research challenges that need to be addressed to bring complex systems within the reach of hybrid optimization techniques.

1 Motivating examples

1.1 Urban Planning

To clarify what is decision making in presence of a complex system, consider a traffic network. Traffic and, more in general, human mobility are key sustainability problems. Currently congestions cost about 50 billion euros per year at the EU level and this figure could rise rapidly if no action is taken to ease the pressure on the road network. Besides the economic aspect, high traffic density has negative effects on the environment in terms of noise and pollutants.

To understand the dynamics of traffic, a number of micro and macro simulators [10] [16] have been developed. Traffic simulators are used to observe the traffic flow of the network, to understand criticalities in terms of congestions, to predict the effect of specific interventions, to forecast traffic burst in real time and possibly to perform an on-line control to improve some figures. Simulators consider the dynamics of cars/pedestrians/public transports, the road infrastructure (traffic lights and roundabouts etc.) and possibly the cognitive aspect of drivers.

Based on the simulator, suppose we want to support a urban planner of a medium-size city (with around 200 junctions). She has a given budget to install traffic light control systems with the aim that the traffic flow is improved and congestions are reduced. There are different types of control systems that differ in price and performances. A combinatorial problem arises as one has to decide how many control systems of each type to buy given a budget envelope and where to place them. In principle, for doing decision making in this context, one has to enumerate and simulate all possible combinations of decision variables assignments that respect budget constraints, i.e., that are feasible.

The combination that has the best impact on traffic (derived from the simulator) is the optimal one. Clearly, this enumerative generate and test behaviour is not practically feasible. Just consider that selecting the junctions requires to chose in a space of 10^{60} combinations. Therefore, traffic experts identify a subset of potential scenarios based on experience and simulate them performing what-if analysis. We could go far beyond what-if analysis by proactively learning from the simulator the impact of decision variable assignments on traffic, learn how connected junctions interact and then insert the extracted knowledge into the optimization and decision model.

In this way, we merge the dynamic aspect of traffic, seen as a complex system, with multiple decision making layers: the legal layer (which rules and norms should be defined) and the urban layer (changes to the road network, addition of traffic lights).

The impact of a system that is able to drive a traffic manager toward optimal solutions is clearly huge. Even a small improvement in congestion reduction and in the efficiency of traffic flow, achieved through a proper collaboration between a simulator and a decision and optimization component, would have a huge beneficial economic and environmental impact and would improve the quality of life in urban environments.

1.2 Energy policy making

As another example, consider a regional energy planner who has to decide the energy share for the region in five years. The energy share depends on the specific energy sources considered and on a set of objectives, strategies and constraints that should be taken into account while planning. For example the region should produce a specific expected outcome of energy within a given budget (public plus private) constraint. The produced energy should come from a balanced energy source diversification and each source has its own limits due to the regional geophysical characteristics. For instance, some regions are particularly windy, while some others are not. Hydroelectric power plants can be built with a very careful consideration of environmental impacts, the most obvious being the flooding of vast areas of land. This poses constraints on the maximum energy that can be produced by a given energy source. Finally environmental pressures should be taken into account while planning as some aspects might be particularly important in a specific region. So basically the policy maker has to solve a combinatorial optimization problem to define a regional energy plan.

As the plants for producing the energy are not built by the region, but are built, managed and maintained by citizens (consider for example photovoltaic plants) or by small enterprises (consider for example biomass power plants) the question is: how a policy maker can decide the proper incentive mechanisms to boost the development of energy production plants. The amount of economic incentives, as well as the way (the mechanism) such incentives are distributed to private stakeholders comes out from the dynamic of complex systems, i.e. the society and the energy market. One might think this relation is linear: the higher the incentives the higher the energy installed. This would be the case is agents are perfectly rational and follows in their decisions only economic aspects. Indeed, agents are driven also by social and cognitive aspects such as belongingness to a group, freedom from the power suppliers, trust in future and government, perceived bureaucracy and these aspects depend also on the number of contacts of each agent.

Having a social simulator one could perform a trial-and.error process to understand the proper incentive level and the mechanism of their distribution that is feasible with a given budget (public) but this may require a long and error prone process.

Therefore a smarter form of integration between simulation and optimization is required.

2 Methodological needs

To enable decision making and optimization techniques to decide on complex systems, we need a methodological paradigm shift and a multi-disciplinary approach to be combined with Constraint programming and more in general hybrid optimization techniques. The traditional interaction between an optimization modeling expert and a domain expert that converges to the classical decision making model should be maintained and enriched via the exploitation of knowledge coming from a simulator that reproduces the temporal dynamic of the complex system it is describing. From the simulator, significant data should be extracted in terms of observables, i.e., significant measures and global data for the modeling component.

The processing of the extracted data should go through three steps devoted to the creation of (part of) a decision (and optimization) model, the extended model in Figure 1. The learning step can be used to understand the relations between the decision variables of the model with their impact on the simulated system. Beside the learning step, an interaction mechanism between the simulator and the decision making model should be designed. The interaction could go from a simple iterative process converging after a number of iterations, to a more sophisticated interaction based on game theory and mechanism design aimed at achieving equilibrium points instead of global optima. Finally, abstraction is needed to cast the learned relations and the knowledge coming from the simulation into extended model components that can be added to the traditional expert-extracted model.

Beside the extraction of an extended model taking into account the effects of decisions on the simulated complex system, there is need to push hybrid decision

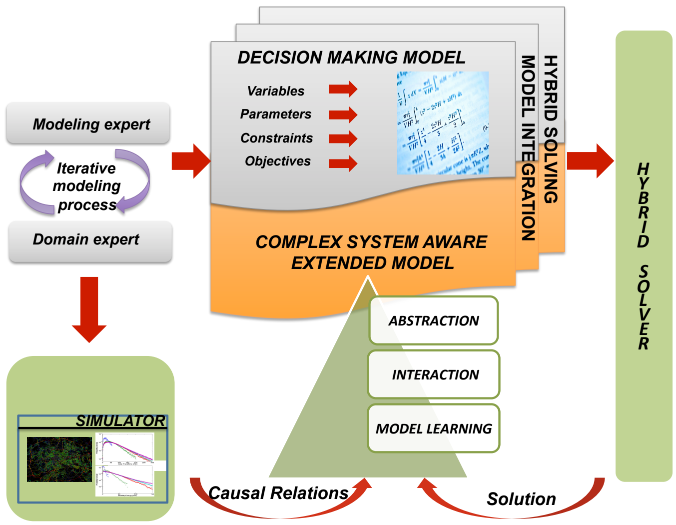

Fig. 1. Structure of the decision making process

making and optimization far beyond the current state of the art. The domains exhibiting a complex system behaviour necessarily present a ultra large scale, due to the huge number of interdependent decision variables, involve multiple, typically conflicting, objectives, stochastic and incomplete data and decision effects, and problem variations over time. This implies that, despite a good level of maturity has been achieved by existing decision and optimization techniques, we need a paradigm shift in the modeling and solving methodology for large, data-driven, multi-scale problems arising from the connection with complex systems.

In addition, the extensive integration of various paradigms from computer science (learning, game theory and decision making/optimization techniques) and physics (dynamic simulation) is one of the most important innovative aspect that needs to be investigated.

2.1 State of the Art

The need to connect mathematical modeling and simulation with optimization has already been recognized in the past. For example the so-called simulation optimization field [21] uses optimization to aid simulation for choosing optimal simulation parameters [9] to improve operations. The goal of optimization routines is to seek improved settings of system parameters with respect to the performance metrics. Similarly, Neuro-Dynamic Programming NDP [7] 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 state space. Markov decision processes have also been used in Reinforcement Learning (RL) together with simulation for learning directly the policy parameters. Both NDP and RL are concerned with how a single agent ought to take actions in an environment so as to maximize some notion of cumulative reward.

On the contrary we are interested in understanding how global political actions and interventions impact on a complex systems without changing the agent behavior. Rather the aim of extended hybrid optimization techniques is to (1) observe/learn the causal link between agents behavior and high level decisions (2) cast these relations into a model component and (3) solve them through decision making techniques.

Coming back to the mobility example, we are interested in understanding which is the best urban plan for reducing congestions by considering economic, environmental and social aspects. The impact of each urban plan on mobility can be extracted from the simulator. The aim is neither to change the drivers behavior nor to enhance their decision making capability, but rather to observe/learn the relations between their behavior and the decision taken by a urban planner and cast these relations into a model component.

Simulation-aware optimization has been considered in the context of Genetic Algorithms [30]. The basic integration technique consists in solving a numerical model through a simulator to evaluate the fitness function. More recent works adopt the same approach for ozone control [15], based on a numerical air quality model and for aerodynamic design of aircrafts [28] minimizing their sonic boom signature at the ground, evaluated via a fluid dynamics simulator. 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 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 [13] 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 [14], a system that integrates in a closed loop simulation and metaheuristics to achieve a 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 our previous work [31], simulation has been used to assess the accuracy of the decision making and optimization model designed by the expert and to measure the difference between the solution quality computed by the optimizer and the simulated one.

We advocate instead a novel, integrated methodology for coupling simulation with optimization and decision making by extracting proper model components and designing a solving process that is aware of the complex system that is the subject of simulation. This result could be achieved by using machine learning techniques to extract a model from simulation-derived data, game theory and mechanism design to find equilibrium points and abstraction to cast the learnt pieces of knowledge into proper model components that can be exploited by a combinatorial decision making and optimization.

Although some approaches in the literature consider aspects related to model learning (either entire models [1], or constraints structures from examples [5]), and to the use of game theory together with optimization (in optimizing data centers for achieving a given quality of service [23]), much work needs to be done to generalize these conclusions.

What we need is a methodology for learning decision making components from simulated complex systems, merge these component with traditional models and solve them efficiently. We envision two mainstream research topics: (1) connecting simulation and decision making through machine learning and game theory (2) pushing decision making and optimization techniques at a ultra large scale.

3 Machine learning interaction

One possible approach for integrating optimization and simulation is to view causal relations between observables and decisions through simulation-extracted data as examples for a learning component. It is very important to note that we have to learn very specific relations that do not describe the temporal dynamic of the system, but instead the link between decision variables assignments in the combinatorial model with simulation-extracted observables.

Coming back to the traffic example, we are not interested in knowing that in one hour the eastbound part of the city will be congested. We are interested in knowing which urban intervention could lower the number of congestions or which laws and norms could improve the traffic flow.

The learnt piece of knowledge can be any component of a decision making process: it can be a constraint (or a set of constraints) [3] and [4], an objective function [14], a cost matrix (in general a set of parameters).

Therefore in this approach the simulator is used to generate a training and a test set. This is done by providing the simulator with a set of assignments (examples, e.g. the same used for the what-if analysis) and the corresponding simulated observable (concept value). Machine learning is applied to extract (part of) the decision making model from the training set and to check its accuracy on the test set. Clearly the language in which we express examples and concepts could influence the performances of the learner.

Note that the machine-learning could be applied even if we do not have access to the simulator that can be considered as a black-box. The only information needed from the simulator is a set of observables that are produces as output of the simulation.

Some open problems should be addressed in this context. First the definition of the training set is crucial for effectiveness of the learning process. Therefore, we could devise some methods to use the optimizer as a guidance for the training set generation. Second, the accuracy of the learnt piece of knowledge is extremely important in the decision making process. Therefore, we should devise techniques that take into account the accuracy of the model while solving the problem.

3.1 Benders Decomposition-like integration

A possible way to integrate a simulator and a decision making component is through a scheme that resembles Benders Decomposition.

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 decision scenario 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 and optimization solver. In principle, one would like to simulate only the best values from the decision making viewpoint; unluckily these values are unknown and depend on the constraints provided by the machine learning component. To improve the interaction, the simulator and the decision making component could be integrated in a Benders-like schema.

The interaction starts from the decision making component, that generates an optimal solution of the master problem. The solution contains tentative decisions that are passed to the simulator, that executes a number of simulations for those values, and provides the correspondent statistics. These statistics might confirm or not the tentative values proposed by the decision making component. In other words, the simulator checks the feasibility of the proposed set of decisions. If the decision scenario is feasible, the iteration stops and the result is provably optimal [6]. On the contrary, if the tentative decisions are not feasible for the simulator, another iteration is required. So, a constraint (or *nogood*) is communicated from the simulator to the decision making model that provides new tentative values to the simulator.

Two are the challenging aspects of this interaction: the first concerns devising the a set of effective no-goods coming from the simulator: 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. The second challenging point is that for Benders-like schema to be effective one has to insert a subproblem relaxation into the master problem. This relaxation could be extracted by using machine learning as explained in section 3, but then we have to be sure that the inferred constraints represent a relaxation and if not, the decision process should be aware of that.

4 Game theory based interaction

Very often, in the real world, optimal points are not interesting. What is interesting instead is an equilibrium point, e.g, Nash equilibrium points. Here is where game theory comes into play: a group of players is in Nash equilibrium if each one is making the best decision that he or she can, by taking into account the decisions of the others. Nash equilibrium does not necessarily mean the best payoff for all the players involved; in many cases, all the players might improve their payoffs if they could somehow agree on strategies different from the Nash equilibrium. We foresee an interaction between the decision making component and the agents composing the simulator [17], as they represent different decision levels and competing actors.

For this reason, the simulation and the decision making components may want to exchange solutions, pay offs, costs and objectives to reach equilibrium points instead of optimal points that are not stable.

An important point is to have truthful evaluation from agents considering how they evaluate specific decision making choices. For example, consider a policy maker who has to define the incentive strategy for pushing the use of renewable energy sources. She should be aware of the real willingness of agents to install renewable energy plants. For example if non truthful mechanisms are considered one agent could say he needs the 20% of his investment to build the plant even if he would have built it anyway. We could devise mechanisms to extract from the simulator an uncertain payoff matrix.

Note that, differently from the machine learning interaction, game theory needs to have access to the simulator, namely to the single agents. Thus, this form of interaction really merges the global decision level and the individual layer.

5 Ultra large scale hybrid decision making

The scale and the complexity of complex systems-related problems is extremely high as they merge various decision levels, present uncertain, probabilistic and incomplete data and often involve conflicting quality measures. The very nature of these problems and their complex structure calls for hybrid decision support and optimization tools. In fact different aspects, criteria, or components of a specific problem cannot be tackled effectively and efficiently by a single solution technique alone. However, solving such complex problems requires the hybridization and composition of different solvers at an unprecedented scale.

State of the art hybrid solvers hardly merge more than two approaches. Surveying the optimization and decision support literature [26], many researchers report success stories on piecewise integration of techniques from Artificial Intelligence, Operations Research and Computational Logics. For example constraint programming (CP) has been successfully integrated with integer linear programming (ILP) [25] in various ways: one possibility is to use problem decomposition, such as Logic Based Benders Decomposition [18], where the ILP and the CP solvers face different problem components. Another decomposition technique for the ILP/CP integration is the CP-based column generation [19]. The strengths of decomposition-based hybrids are that (1) they are proved to converge to the optimal solution and (2) they effectively face structured sub-problems: each of them is treated separately with the most suitable modeling and solving technique. Another form of piecewise integration considers enriching tree search of constraint programming with local search (LS) and meta-heuristics (MH) [12]. Also Local search and Integer Linear programming have been recently integrated into the Local Branching framework [11]. Recently, SAT encodings have been used in constraint programming global constraints [29].

A disadvantage of these hybrids is that they are often specifically tailored for a particular application requiring a complete redesign and reimplementation when a new problem is to be tackled and that they usually combine only two techniques.

When coping with decision making in presence of complex systems, larger and more structured problem come into play, with many sub-problems interacting in a non-trivial, stochastic, and dynamic manner. We therefore need a methodology for problem solving component hybridization, composition and orchestration that enables the integration of more than two components and that might take into account also multi-disciplinary components such as learners and game theoretical methods into the problem solving architecture.

References

1. Lallouet A, M. Lopez, L. Martin, and C. Vrain. On learning constraint problems. In *Proc. of ICTAI*, 2010.
2. P.W. Anderson. Physics: The opening to complexity? In *Proceedings of the National Academic Science*, 1995.
3. A. Bartolini, M. Lombardi, M. Milano, and L. Benini. Neural constraint for solving real world problems. In *Proc. of CP 2011*, 2011.
4. A. Bartolini, M. Lombardi, M. Milano, and L. Benini. Optimization and controlled systems: A case study on thermal aware workload dispatching. In *Proc. of AIII 2012*, 2012.
5. N. Beldiceanu and H. Simonis. A constraint seeker: Finding and ranking global constraints from examples. In *Proc. of CP 2011*, 2011.
6. J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.
7. D.P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
8. C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 2003.

9. G. Deng. *Simulation-Based Optimization.* PhD thesis, University of Wisconsin Madison, 2007.
10. EURECOM. Survey on mobility models for vehicular ad hoc networks: A survey and taxonomy. http://www.eurecom.fr/util/publidownload.fr.htm?id=1951.
11. M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98, 2003.
12. F. Focacci and F. Laburthe andA. Lodi. Local search and constraint programming: Ls and cp illustrated on a transportation problem. In *Constraint and Integer Programming.* Kluwer, 2003.
13. M.C. Fu. Optimization via simulation: A review. *Annals of Operations Research*, 53, 1994.
14. F. Glover, J.P. Kelly, and M. Laguna. New advances for wedding optimization and simulation. In *Proc. of the Winter Simulation Conference*, 1999.
15. Loughlin D. H., S. R. Ranjithan, Jr. J. W. Baugh, and E. D. Brill Jr. Application of gas for the design of ozone control strategies. *J. Air Waste Manage. Assoc.*, 50, 2000.
16. D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Phisics*, 73, 2001.
17. A. Holland and B. O'Sullivan. Survey of game theoretic tools in dynamic environments for policy management. Technical report, 4C, 2012. ePolicy Project Deliverable 5.1.
18. J. N. Hooker. Logic-based benders decomposition. *Mathematical Programming*, 96, 2003.
19. U. Junker, S. E. Karisch, N. Kohl, B. Vaaben, T. Fahle, and M. Sellmann. A framework for constraint programming based column generation. In *Proc. of CP1999*, 1999.
20. K. Kaneco. *Life: An Introduction to Complex Systems Biology.* Springer, 2006.
21. A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis.* Mc Graw Hill, 2000.
22. M. Levy, H. Levy, and S. Solomon. *Microscopic Simulation of Financial Markets; From Investor Behaviour To Market Phenomena.* Academic Press, 2000.
23. B. Lubin, J.O. Kephart, Das R, and D.C. Parkes. Expressive power-based resource allocation for data centers. In *Proc. of IJCAI 2009*, 2009.
24. K. Marriott and P. Stuckey. *Programming with Constraints: An Introduction.* MIT Press, 1998.
25. M. Milano. *Constraint and Integer Programming: toward a unified methodology.* Kluwer, 2003.
26. M. Milano and P. Van Hentenryck. *Hybrid Optimization: the ten years of CPAIOR.* Springer, 2010.
27. G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization.* Wiley Interscience Series in Discrete Mathematics and Optimization, 1988.
28. S.and D. Sasaki Obayashi, Y. Takeguchi, and N. Hirose. Multiobjective evolutionary computation for supersonic wing-shape optimization. *IEEE Trans. Evol. Comput*, 4, 2000.
29. O. Ohrimenko, P.J. Stuckey, and M. Codish. Propagation via lazy clause generation. *Constraints*, 14(3), 2009.
30. Haupt R.L. and Haupt S.E. *Practical Genetic Algorithms.* Wiley, 2004.
31. M. Ruggiero, A. Guerri, D. Bertozzi, M. Milano, and L. Benini. A fast and accurate technique for mapping parallel applications on stream-oriented mpsoc platforms with communication awareness. *International Journal of Parallel Programming*, 36(1)(3-36), 2010.

32. D.M.D. Smith and N. F. Johnson. Predictability, risk and online management in a complex system of adaptive agents. *Arxiv preprint physics*, 0605065, 2006.
33. Y. Bar Yam. *Dynamic of Complex Systems*. Addison Wesley, 1997.