Modeling the Management of Water Resources Systems Using Multi-Objective DCOPs

Francesco Amigoni
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
francesco.amigoni@polimi.it

Andrea Castelletti
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
andrea.castelletti@polimi.it

Matteo Giuliani
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
matteo.giuliani@polimi.it

ABSTRACT

Multiagent systems are being increasingly used in environmental modeling applications to characterize human behavior and interactions with natural processes. A model based on Distributed Constraint Optimization Problems (DCOPs) has been recently proposed for studying the management of water resources systems from the point of view of a regulating institution in charge of coordinating multiple distributed decision-makers (agents). However, this DCOP-based model does not explicitly account for the variety of stakeholders' and regulators' interests that are generally involved, representing incommensurable and often competing objectives. In this paper, we provide a Multi-Objective DCOP (MO-DCOP) model that supports distributed water resources management through the exploration of tradeoffs across different agents' objectives. Among the available algorithms for solving the resulting MO-DCOP, we choose a variant of the B-MOMS algorithm because it allows identifying (an approximation of) the whole Pareto frontier for the problem. Experimental results conducted on a number of randomly generated water systems show that the approximation introduced by the solving algorithm is limited for most of the systems.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms, Experimentation, Management

Keywords

Water resources systems; Multi-objective DCOPs; Pareto frontier

1. INTRODUCTION

Approaches based on multiagent systems (MASs) are increasingly used in modeling environmental systems because of their advantages over traditional centralized strategies, particularly in evaluating macro-level properties emerging from lower-level interactions among the agents [3]. In addition, agent-based models provide a more natural description of some distinguishing features of environmental systems: MASs can (*i*) deal with spatial variability, (*ii*) represent a population of heterogeneous individuals, (*iii*) explicitly model multiple, distributed, and autonomous decision-makers. Focusing on the management of water resources systems, an agent approach has been recently proposed for studying the design of coordination mechanisms among multiple distributed decision-makers (agents) [10]. Specifically, the work provides two formulations of the problem from the standpoint of the water authority in charge of regulating the system. In the first one, formalized as a Distributed Constraint Satisfaction Problem (DCSP) [23, Chapter 1], the relevant stakeholders are considered making decisions about the amount of water they withdraw or release in the system only subject to physical constraints that cannot be violated. However, their individualistic decisions can produce negative externalities that strongly impact on the system-wide benefit. In the second one, formulated as a Distributed Constraint Optimization Problem (DCOP) [24, Chapter 12], the regulating institution introduces normative constraints, whose violation has a cost, in order to increase the global benefit. Both formulations of [10] do not explicitly account for the incommensurable, and often competing, objectives of the different stakeholders.

In this paper, we propose a *Multi-Objective DCOP* (MO-DCOP) formulation which extends the work of [10] by allowing the exploration of the tradeoffs among the agents' objectives. This multi-objective point of view helps the regulating authority in performing scenario analysis to reduce the negative decision biases from both cognitive myopia, where narrow or restrictive definitions of optimality strongly limit the discovery of relevant decision alternatives, and cognitive hysteresis, where traditional strategies for addressing a problem restrict the generation of new hypotheses for innovative decisions or additional objectives [5].

With our model, we make a step in this direction by providing a way to calculate the Pareto frontier of an MO-DCOP, where objectives and constraints are represented as vectors of functions independently optimized and no strategic behavior of the agents is considered. Among the available algorithms for solving MO-DCOPs, we selected B-MOMS [6] because it easily allows to calculate the set of solutions that are on the Pareto frontier, starting from an acyclic *factor*

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

graph, which represents the structure of the underlying MO-DCOP problem. In this work, we introduce a variant in the original B-MOMS algorithm and a number of techniques to handle cyclic factor graphs, as many water resources systems' representations are actually cyclic.

In summary, the main contributions of this paper are the following. (i) An MO-DCOP representation for the management of water resources systems. This specific class of MO-DCOPs is characterized by having all constraints (except one) containing a single objective function. (ii) The proposal and experimental evaluation, in the context of our application, of a number of methods for making the factor graph representations of MO-DCOPs acyclic. (iii) The *MO-Max-Sum* method for finding the Pareto frontier of the acyclic version of the problem that operates as a variant of B-MOMS and that is experimentally compared against the Pareto frontier for the original problem found by a brute-force algorithm.

2. RELATED WORK

MAS approaches have become widely used in several environmental modeling contexts [2]. The primary goal of most of these studies, also referred to as multiagent simulations (for a review, see [1,4] and references therein), is to simulate complex systems in order to evaluate macro-level properties emerging from lower-level interactions among the agents. Purely reactive agents are largely adopted as a modeling approach to define behavioral rules which react to environmental changes (e.g., [12,14,22]). However, the prescriptive use of MAS models in decision support systems remains a challenge due to their mathematical complexity, which requires to shift toward a descriptive standpoint [7], developing what-if analyses with respect to a limited number of management alternatives and modeling simple decision mechanisms based on linear programming [21]. In the water resources literature, the first contribution adopting proactive MAS for a non-dynamic optimization problem was presented in [26] and further developed by [10]. A similar approach was then adopted to optimize pre-season farmers decisions [18] and to simulate optimization-driven water markets [11] and emission trading [19]. [9] proposed an agent-based optimization framework to assess the value of cooperation in large-scale transboundary water resources systems.

The most significant previous work for this paper is that of [10], whose goal is the design of a coordination strategy for multiple distributed decision-makers acting in the same water system. Three solutions are comparatively analyzed: an uncoordinated solution in which all the agents act independently, a coordinated solution in which the decisions of the agents are only physically constrained (e.g., water availability, canals capacity), and a coordinated solution in which also normative constraints are taken into account. The last two solutions are modeled as DCSP and DCOP, respectively. The results of numerical simulations show that the DCSP and DCOP formulations can better approximate the ideally-optimal solution calculated in a centralized fashion, which represents the solution that guarantees the maximum of the system-wide benefit. An important issue missing from the above study is the fact that the multi-objective aspects of the problem have not been considered. In the DCOP formulation, the agents' objective functions are aggregated (summed up) together with constraints, preventing the regulator from knowing the Pareto frontier of the problem.

In this paper, we provide an MO-DCOP formulation of the case study of [10], which can be easily applied to other water systems, as we show in our experiments. In order to solve the MO-DCOP, we used the Bounded Multi-Objective Max-Sum (B-MOMS) algorithm [6], which is structured in three phases:

- *Bounding*, which obtains an acyclic factor graph representation of the problem. The idea is that of building a maximal spanning tree by weighting the edges of the original factor graph.
- *Max-Sum*, during which the Pareto-optimal solutions of the acyclic problem obtained in the previous phase are calculated by a generalization to the multi-objective case of the Bounded Max-Sum algorithm [20].
- *Value-propagation*, in which the agents select a Pareto optimal assignment for their variables.

Theorem 1 of [6] proves that the set of solutions obtained after the Max-Sum phase are Pareto optimal for acyclic factor graphs. Moreover, the authors provide a bound over the approximation of the solutions returned by the B-MOMS algorithm with respect to the original (cyclic) problem. We exploit the first result to calculate the set of non-dominated solutions for the acyclic version of our problem and we experimentally evaluate the difference with the Pareto frontier of the original problem.

An alternative approach for solving MO-DCOPs is MO-Adopt [16], which generalizes to the multi-objective case the Adopt algorithm [17] used for solving DCOPs. It is based on a search strategy that exploits pseudo-trees and techniques for pruning branches. The search process is based on two messages: CONTEXT, which informs children of a variable in the tree about the parent's and ancestors' assignments, and COST, which informs the parent variable about the cost relative to the current context. Differently from B-MOMS, MO-Adopt can deal with problems with cyclic factor graph representations. Yet, since it builds a solution incrementally, it can find only a single solution on the Pareto frontier. A naive approach for finding the whole Pareto frontier using MO-Adopt is to iteratively generate solutions by considering the solutions already found as constraints, but the computation cost can become prohibitively expensive.

3. WATER RESOURCES SYSTEMS AS MULTI-OBJECTIVE DCOPS

In this section, we first survey MO-DCOPs and, then, we introduce an example of a water resources system along with its representation as MO-DCOP.

3.1 MO-DCOPs

A *Multi-Objective DCOP* (MO-DCOP) is defined as a problem in which n agents control n variables $\{x_1, x_2, \ldots, x_n\}$ (usually, one variable per agent). Call $\mathbf{x}$ the vector of the n variables. Variables can take values from finite domains $D_1, D_2, \ldots, D_n$, respectively. Assignments of values to variables should maximize a set of k objective functions:

$$\mathbf{F}(\mathbf{x}) = \left[F^1(\mathbf{x}), F^2(\mathbf{x}), \ldots, F^k(\mathbf{x})\right]^T$$

Vector $\mathbf{F}(\mathbf{x})$ is decomposed in a number m of constraints:

Agent	Objective function and constraints	Notation
City	$\max_{x_1} f_1(x_1) = a_1x_1^2 + b_1x_1 + c_1$ subject to $\begin{cases}\alpha_1 - x_1 \le 0 \\ \alpha_2 - Q_1 + x_1 \le 0\end{cases}$	x_1: water withdraw for municipal use Q_1: mainstream inflow α_1: minimum water demand for the city α_2: minimum water requirement for the dam
Farm$_1$	$\max_{x_4} f_4(x_4) = a_4x_4^2 + b_4x_4 + c_4$ subject to $\begin{cases}\alpha_3 - x_4 \le 0 \\ \alpha_4 - Q_2 + x_4 \le 0\end{cases}$	x_4: water withdraw for irrigation Q_2: tributary inflow α_3: minimum water demand for the farm α_4: minimum water requirement for the tributary
Farm$_2$	$\max_{x_6} f_6(x_6) = a_6x_6^2 + b_6x_6 + c_6$ subject to $\begin{cases}\alpha_5 - x_6 \le 0 \\ \alpha_6 - x_2 - x_3 + x_6 \le 0\end{cases}$	x_6: water withdraw for irrigation α_5: minimum water demand for the farm α_6: minimum water requirement for the mainstream
Dam	$\max_{x_2} f_2(x_2) = a_2x_2^2 + b_2x_2 + c_2$ subject to $x_2 - S - Q_1 + x_1 \le 0$	x_2: water release S: reservoir storage
Eco$_1$	$\max_{x_3} f_3(x_3) = a_3x_3^2 + b_3x_3 + c_3$ subject to $\begin{cases}x_3 = Q_2 - x_4 \\ \alpha_4 - x_3 \le 0\end{cases}$	x_3: water flow through the tributary
Eco$_2$	$\max_{x_5} f_5(x_5) = a_5x_5^2 + b_5x_5 + c_5$ subject to $\begin{cases}x_5 = x_2 + x_3 - x_6 \\ \alpha_6 - x_5 \le 0\end{cases}$	x_5: water flow through the lower mainstream

Table 1: The agents of the example water resources system

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{m} \mathbf{F}_i(\mathbf{x}_i)$$

where each constraint i is a vector of k objective functions:

$$\mathbf{F}_i(\mathbf{x}_i) = \left[F_i^1(\mathbf{x}_i), F_i^2(\mathbf{x}_i), \ldots, F_i^k(\mathbf{x}_i)\right]^T$$

and $\mathbf{x}_i \subseteq \mathbf{x}$ is the set of variables (agents) involved in the constraint i.

A MO-DCOP can be represented by a *factor graph*, like the one shown in Figure 1, where nodes represent variables and constraints (variable nodes and function nodes, respectively) and edges represent involvement of variables in constraints.

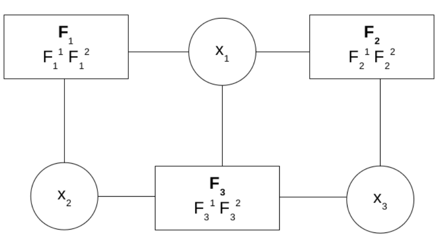

Figure 1: An example of a multi-objective factor graph with 3 variable nodes x_1, x_2, x_3 and 3 function nodes F_1, F_2, F_3 (the two objective functions are $F^1 = F_1^1 + F_2^1 + F_3^1$ and $F^2 = F_1^2 + F_2^2 + F_3^2$)

The solution of a MO-DCOP is the set of *Pareto optimal* alternatives, namely the assignments $\mathbf{x}$ that are not dominated. An assignment $\mathbf{x}$ is dominated when there is another assignment $\mathbf{x}'$ such that $F^j(\mathbf{x}') \geq F^j(\mathbf{x})$ for all $j \in [1, k]$ and $F^{j'}(\mathbf{x}') > F^j(\mathbf{x})$ for at least a $j' \in [1, k]$. A trivial upper bound on the number of Pareto optimal solutions is $\prod_{i=1}^{n} |D_i|$.

3.2 An example of a water resources system

We now illustrate a simple synthetic case study, firstly introduced by [26], to describe how to model water resources systems as MO-DCOPs and to test the algorithm we propose for calculating the Pareto frontier of the problem.

The system, see Figure 2, is composed of an hypothetical Y-shaped river, with a city and a water reservoir, operated for hydropower generation, located on the mainstream branch, a farm on the bottom tributary, and another farm downstream of the rivers junction. Two ecological points of interest, primarily for the protection of fish habitats, are also present. We consider the perspective of the regulating institution in charge of managing the water system and balancing all the different objectives associated to the relevant stakeholders.

This non-dynamic problem, although its apparent simplicity, caters for multiple sources of complexity characterizing many real world applications: upstream-downstream power asymmetry, as in the Nile River Basin [25]; the presence of agents making decisions in parallel and in series, as in the Yellow River Basin [27]; the difference between primary objectives associated to real decisions (e.g., water supply demands driving the amount of water to divert from the river or hydropower production determining the releases from the dam) and secondary environmental concerns, as in the Zambezi River Basin [9].

Each stakeholder (cities, dams, farms, ecological protection) can be represented by an agent i, that controls its variable x_i, which is the amount of water withdrawn from the river or released from the reservoir, according to an objective function f_i and to some constraints, see Tables 1 and 2. City, Farm$_1$, and Farm$_2$ agents decide the amount of water they withdraw to satisfy domestic or irrigation water demand (x_1, x_4, and x_6, respectively), Dam agent decides on the amount of water to release for hydropower generation (x_2), while Eco_1 and Eco_2 agents are passive and can deal only with decisions made by other agents. Agents make decisions for maximizing their objectives. A quadratic concave objective function is assigned to each agent to preserve the nonlinear complexity of real objective functions. Constraints are either hard or soft. *Hard constraints* refer to physical constraints that are enforced by nature; for example, a hard constraint can state that the amount of water withdrawn from a river cannot be larger than the amount of water present in that river. Hard constraints cannot be violated by any solution. *Soft constraints* are normative constraints that can be enforced by the managing institution; for example, a soft constraint can state the minimum amount of water that should be left in a river after withdrawn for municipal use. Soft constraints can be violated by a solution incurring in a cost.

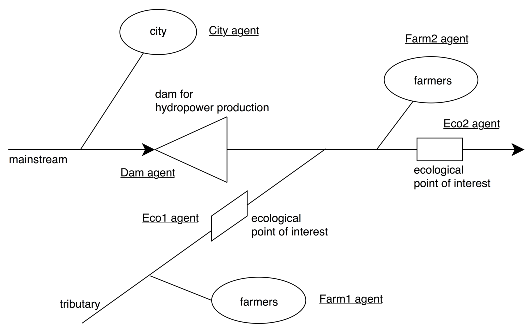

Figure 2: A synthetic water resources system

i	a_i	b_i	c_i	α_i
1	−0.2	6	−5	12
2	−0.06	2.52	0	10
3	−0.29	6.38	−3	8
4	−0.13	5.98	−6	6
5	−0.055	3.63	−23	15
6	−0.15	7.5	−15	10

Flow scenario	Q_1	Q_2	S
High flow	80	35	10
Medium flow	40	20	8
Low flow	15	8	3

Table 2: Values for parameters (left) and flow scenarios (right) of Table 1

3.3 Modeling the water resources system as MO-DCOP

The above example water resources system can be modeled as a MO-DCOP as follows.

- n agents $\{1, 2, \ldots, n\}$, corresponding to the active agents (hence, excluding Eco_1 and Eco_2 in our example), controlling n variables $\{x_1, x_2, \ldots, x_n\}$, where agent i controls variable x_i.
- n domains $D_1, D_2, \ldots, D_n$, which are finite and contain positive values. In particular, each D_i contains values around $\hat{x}_i$, which yields the maximum of f_i.
- n (local) objective functions $F^{Oi}(x_i) = f_i(x_i)$, one for each agent i.
- A single (global) objective function $F^G = \sum F_j$, which aggregates all the constraints, hard and soft.

A water resources system composed of n active agents is then modeled as an MO-DCOP that involves $n+1$ objectives:

$$\mathbf{F}(\mathbf{x}) = \left[F^{O1}(\mathbf{x}), F^{O2}(\mathbf{x}), \ldots, F^{On}(\mathbf{x}), F^{G}(\mathbf{x})\right]^T$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is the vector of the variables.

Since in an MO-DCOP all utility functions are referred to constraints, the local objective functions $F^{Oi}(x_i)$ are modeled as unary constraints involving only the variable x_i. The hard and soft constraints of the water system problem are all aggregated in F^G. As consequence, we are facing a very particular version of MO-DCOP in which each constraint involves exactly a single objective function:

$$\mathbf{F}_i(\mathbf{x}_i) = [0, \ldots, f_i(x_i), \ldots, 0]^T \quad \forall i \in [1, n]$$

$$\mathbf{F}_{n+1}(\mathbf{x}) = \left[0, \ldots, 0, F^G(\mathbf{x})\right]^T$$

Our test case is formulated with $f_i(x_i)$s reported in Table 1 and the constraints F_j in Table 3 aggregated in F^G. The hard (physical) constraints are represented as constraints whose violation has utility $-\infty$. For example, the fact that an agent cannot withdraw more water than that in the river can be modeled as in the constraint F_2 in Table 3. The soft (normative) constraints of the water system are represented as constraints whose violation decreases their utility. For example, the minimum amount of water that an agent should leave in a river can be expressed as constraint F_1 of Table 3.

In order to allow the institution managing the water resources system to identify appropriate coordination strategies, *all* the non-dominated solutions, namely the *Pareto frontier*, of the above MO-DCOP should be derived. In the next section, we propose a decentralized approach for addressing this issue.

4. THE PROPOSED ALGORITHM

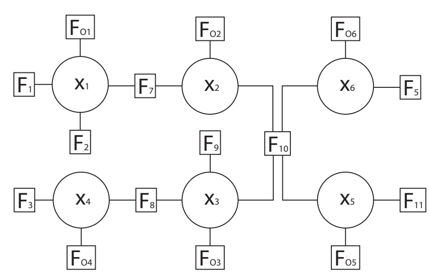

Figure 3: Factor graph for the example water resources system

Given the test case formulated as above, the corresponding factor graph is shown in Figure 3. To run the proposed MO-Max-Sum algorithm, a variant of the B-MOMS one [6], we need an acyclic constraint graph. Since, in general, factor graphs may contain cycles, we make the factor graph acyclic by cutting edges. For example, cutting the edge between x_j and $F_i(\mathbf{x}_i)$ means considering the new function $F'_i(\mathbf{x}_i \backslash x_j)$ (where $\mathbf{x}_i \backslash x_j$ is the vector $\mathbf{x}_i$ without x_j). This introduces an approximation and, specifically, the solutions found by the algorithms are not always guaranteed to be

Constraint	Objective function
$\alpha_1 - x_1 \leq 0$	$F_1(x_1) = \begin{cases} 0 & \alpha_1 - x_1 \leq 0 \\ x_1 - \alpha_1 & \text{otherwise} \end{cases}$
$\alpha_2 - Q_1 - x_1 \leq 0$	$F_2(x_1) = \begin{cases} 0 & \alpha_2 - Q_1 - x_1 \leq 0 \\ Q_1 - x_1 - \alpha_2 & 0 \leq Q_1 - x_1 \leq \alpha_2 \\ -\infty & \text{otherwise} \end{cases}$
$\alpha_3 - x_4 \leq 0$	$F_3(x_4) = \begin{cases} 0 & \alpha_3 - x_4 \leq 0 \\ x_4 - \alpha_3 & \text{otherwise} \end{cases}$
$\alpha_4 - Q_2 + x_4 \leq 0$	$F_4(x_4) = \begin{cases} 0 & \alpha_4 - Q_2 + x_4 \leq 0 \\ Q_2 - x_4 - \alpha_4 & 0 \leq Q_2 - x_4 \leq \alpha_4 \\ -\infty & \text{otherwise} \end{cases}$
$\alpha_5 - x_6 \leq 0$	$F_5(x_6) = \begin{cases} 0 & \alpha_5 - x_6 \leq 0 \\ x_6 - \alpha_5 & \text{otherwise} \end{cases}$
$\alpha_6 - x_2 - x_3 + x_6 \leq 0$	$F_6(x_2, x_3, x_6) = \begin{cases} 0 & \alpha_6 - x_2 - x_3 + x_6 \leq 0 \\ x_2 + x_3 - x_6 - \alpha_6 & \text{otherwise} \end{cases}$
$x_2 - S - Q_1 + x_1 \leq 0$	$F_7(x_1, x_2) = \begin{cases} 0 & x_2 - S - Q_1 + x_1 \leq 0 \\ -\infty & \text{otherwise} \end{cases}$
$x_3 = Q_2 - x_4$	$F_8(x_3, x_4) = \begin{cases} 0 & x_3 = Q_2 - x_4 \\ -\infty & \text{otherwise} \end{cases}$
$\alpha_4 - x_3 \leq 0$	$F_9(x_3) = \begin{cases} \alpha_4 - x_3 \leq 0 & \\ x_3 - \alpha_4 & \text{otherwise} \end{cases}$
$x_5 = x_2 + x_3 - x_6$	$F_{10}(x_2, x_3, x_5, x_6) = \begin{cases} 0 & x_5 = x_2 + x_3 - x_6 \\ -\infty & \text{otherwise} \end{cases}$
$\alpha_6 - x_5 \leq 0$	$F_{11}(x_5) = \begin{cases} 0 & \alpha_6 - x_5 \leq 0 \\ x_5 - \alpha_6 & \text{otherwise} \end{cases}$

Table 3: Constraint functions aggregated in F^G

on the Pareto frontier. Authors in [6] present a bound over the amount of this approximation based on the cuts operated to obtain an acyclic graph. Unfortunately, this bound cannot be significantly applied to our problem because the constraints that introduce cycles in our factor graph are hard constraints and their worst value is $-\infty$. We hence resort to the experimental evaluation to assess the effects of this approximation.

4.1 Building acyclic factor graphs

To make the factor graph acyclic we can use a maximum spanning tree algorithm (as in [6]) that builds a tree by cutting edges in the original factor graph in order to maximize the weights of the edges in the tree. The problem is reduced to the Minimum Spanning Tree (MST) problem by negating the weights and can be solved by distributed algorithms, like GHS [8], but, for the sake of simplicity, we use Kruskal's centralized algorithm [13]. For MST algorithms, weights should be associated to edges: those with larger weights are less likely to be cut. Given the importance of this operation, we propose and experimentally evaluate (in the next section) different techniques to obtain acyclic factors.

Uniform (U) This is the technique used in [6], when we consider that all our hard constraints have value $-\infty$. In our case, the hard constraints are equally important and are associated the same weight (e.g., 1), while soft constraints have weights 0.

Ranking Nodes (RN) With this technique we set the weight of an edge between x_j and F_i equal to the sum of the degrees (number of incident edges) of x_j and F_i. The rationale is to cut edges less critical.

Ranking Upstream (RU) The idea of this technique is exploit the knowledge about the water system domain and to assign larger weights to the edges connected to upstream variables and smaller weights to the edges connected to downstream variables. The rationale is to preserve upstream edges from which downstream edges and variables depend. The weights starts from 1 for the last variable node and increase by 1 for other variable nodes going upstream. The weight of an edge is equal to the weight of the variable node it connects.

Constraint More (CM) This technique produces a factor graph that is overconstrained with respect to the original factor graph. Considering for example the constraint F_6 of Table 3, cutting the edge between x_6 and F_6 constraints more the values that other variables can assume. CM will prefer to cut these edges by weighting less edges connecting variables like x_6 (more generally, variables with positive sign in constraints of Table 3).

Constraint Less (CL) This techniques is the dual of CM and assigns smaller weights to edges connecting variables with negative sign in the constraints of Table 3 (e.g., the edge between x_2 and F_6).

4.2 MO-Max-Sum algorithm

Once the acyclic factor graph has been obtained, the proposed MO-Max-Sum algorithm operates like B-MOMS but make a slightly different use of the marginal functions of variables in the factor graph. Specifically, in the B-MOMS algorithm the marginal functions are used in the value-propagation phase to select a Pareto-optimal solution. In the proposed MO-Max-Sum algorithm, instead, the value-propagation phase is substituted by a phase that assembles the solutions on the Pareto frontier. The marginal function of a variable node x_j, $z_j(x_j)$, is calculated as a sum of r messages received from the neighboring function nodes $M(j)$:

$$z_j(x_j) = \sum_{i \in M(j)} r_{i \to j}(x_j) = \arg\max_{\mathbf{x} \setminus x_j} \sum_{i=1}^{m(=n+1)} U_i(\mathbf{x}_i) \quad (1)$$

The information stored in the marginal function is, for each value of x_j, the best solutions, in terms of utility, that can be currently obtained. In other words, $z_j(x_j)$ can be thought as a vector that, in correspondence of a value for x_j, stores the best solutions obtainable with that value.

The MO-Max-Sum algorithm is shown in Algorithm 4.1. Variable nodes calculate and send q messages, while function nodes calculate and send r messages. At the beginning of the algorithm, all q messages are initialized to 0, hence the r message sent by a function node F_i to a variable node x_j is a maximization of F_i with respect to all the variables of the scope of F_i except x_j. After this initialization, at each iteration t, each message q is calculated as:

$$q^t_{j\to i}(x_j) = \sum_{k\in M(j)\setminus i} r^{t-1}_{k\to j}(x_j) \qquad (2)$$

using the r messages received by the variable node x_j in the previous iteration $t-1$ from its neighboring function nodes $M(j)$. Similarly, at each iteration t, a function node F_i waits for q messages from its neighboring variable nodes $N(i)$ and calculates its r message as:

$$r^t_{i\to j}(x_j) = \max_{\mathbf{x}_i \setminus\ x_j} \left(F_i(\mathbf{x}_i) + \sum_{k\in N(i)\setminus j} q^t_{k\to i}(x_k) \right) \qquad (3)$$

Note that r messages at t are calculated using the $N(i)-1$ (all the neighboring variable nodes except the one to which the r messages is sent) q messages at t; while q messages at t are calculated using the $M(j)-1$ (all the neighboring function nodes except the one to which the q message is sent) r messages at $t-1$. As far as the above dependencies are satisfied, the order in which the r and q messages are actually sent and received is not important.

Note also that unary function nodes (corresponding to local objective functions) send r messages whose content is the maximization of the corresponding function with respect to its single argument. This message is fixed over iterations and can be calculated only once, slightly limiting the computational effort. It is possible to consider the hard constraints when a function node F_i computes the r message for a variable node x_j. If, for a value of x_j, the corresponding vector violates a hard constraint (i.e., the associated utility is $-\infty$), then the vector is discarded.

If the marginal function calculated by a variable node x_j at iteration t is equal to that calculated by the same variable node at the previous λ iterations, then the marginal function has converged and the variable node is "disabled". When disabled, a variable node x_j will not send messages any longer and the neighboring function nodes will use the last received message from x_j for further calculations. The MO-Max-Sum algorithm terminates when all variable nodes are disabled, namely when all the marginal functions are invariant. In order to assure that all the marginal functions have converged, the value of λ should be set in order to allow the propagation of messages over the factor graph, a safe value being to the maximum distance between two nodes in the factor graph (being it a tree, this distance is twice the depth of the tree).

Algorithm 4.1: MO-MAX-SUM(*variables*, *functions*)

procedure VARIABLE-NODE($M(j)$)
  for each $i \in M(j)$
    do send $q^1_{j\to i}$ message initialized to 0
  for $t \leftarrow 2$ **to** termination
    do {
      for each $i \in M(j)$
        do {
          while there are $r^{t-1}_{i\to j}$ messages unreceived
            do wait for $r^{t-1}_{i\to j}$ messages
          calculate (2) and send $q^t_{j\to i}$ message }
      $z_j(x_j) \leftarrow$ calculate marginal function (1)
      if marginal function is invariant for λ steps
        then stop }
  disable x_j
  return $(z_j(x_j))$

procedure FUNCTION-NODE($N(i)$)
  for $t \leftarrow 1$ **to** termination
    do {
      for each $x_j \in N(i)$
        do {
          if x_j disabled
            then $q^t_{j\to i} \leftarrow q^{t-1}_{j\to i}$ }
      for each $j \in N(i)$
        do {
          while there are $q^t_{j\to i}$ messages unreceived
            do wait for $q^t_{j\to i}$ messages
          calculate (3) and send $r^t_{F_i\to x_j}$ message }
      if all neighboring nodes $N(i)$ are disabled
        then stop }

procedure CALCULATEFRONTIER(*marginalfunctions*)
  frontier $\leftarrow z_1(x_1)$ with $z_1(x_1) \in$ *marginalfunctions*
  marginalfunctions $\leftarrow$ *marginalfunctions*$\setminus z_1(x_1)$
  for each $z_j(x_j) \in$ *marginalfunctions*
    do *frontier* $\leftarrow$ *frontier* $\cap\ z_j(x_j)$
  return (*frontier*)

main
  marginalfunctions $\leftarrow \emptyset$
  for each $x_j \in$ *variables*
    do *marginalfunctions* $\leftarrow$ *marginalfunctions* $\cup$
  $\cup$ VARIABLE-NODE($M(j)$))
  for each $F_i \in$ *functions*
    do FUNCTION-NODE($N(i)$))
  while there are variable nodes not disabled
    do wait
  Paretofrontier $\leftarrow$ CALCULATEFRONTIER(*marginalfunctions*)

After all the variable nodes have been disabled, the marginal functions calculated in the last iteration are intersected to obtain the Pareto frontier. This last step is based on the following proposition.

PROPOSITION 1. *Given a factor graph, a solution* $\mathbf{x}$ *is Pareto optimal if and only if* $\mathbf{x}$ *is present in the marginal functions* $z_j(x_j)$ *of all variables* x_j *of the factor graph, once they have converged.*

PROOF. (Sketch) *Only if part.* A Pareto optimal solution is associated to a non-dominated vector $\mathbf{x}$. The vectors contained in r messages are calculated by the function nodes using a maximization and so they are not dominated. These vectors are propagated in the factor graph in a finite number of steps using q messages. Hence, the sum of non-dominated vectors, contained in r messages, is received by all variable nodes of the factor graph, which add it to their marginal functions. *If part.* If a vector $\mathbf{x}$ is present in all the marginal functions, then there is an assignment in which a value is assigned to each variable such that $\mathbf{x}$ is non-dominated and so it is Pareto optimal. □

Note that it can happen that a marginal function $z_j(x_j)$ contains, in correspondence to a value $\overline{x_j}$, some dominated vectors (meaning that with $x_j = \overline{x_j}$ solutions are obtained that are dominated by those obtained with different values for x_j). However, the MO-Max-Sum algorithm guarantees (through the maximization while calculating the r messages) that these dominated vectors are not propagated and so they cannot belong to the intersection of all marginal functions and, as a consequence, to the Pareto frontier, accordingly to what one might expect.

The worst-case time complexity of the MO-Max-Sum algorithm is dominated by the computation of the r messages as in (3). When all domains have d values, the complexity is $\mathcal{O}(d^{a+2})$, where a is the largest arity (number of arguments) of the F_is. Computing q messages and marginal functions $z_j(x_j)$ has complexity $\mathcal{O}(d)$.

5. EXPERIMENTAL ANALYSIS

5.1 Experiment setting

We developed a generator of realistic water systems in order to heavily test the MO-Max-Sum algorithm. It randomly creates water systems by composing several instances of agents City, Dam, and Farm (Table 1) along a mainstream and some tributaries. Parameters and flow scenarios are also randomly generated to obtain plausible values like those of Table 2.

Given the cuts operated to the factor graph to make it acyclic, some solutions found by MO-Max-Sum could be unfeasible for the original problem. Call $\mathcal{S}_F$ the set of feasible solutions and $\mathcal{S}_U$ the set of unfeasible solutions, the set of all solutions found by MO-Max-Sum is $\mathcal{S} = \mathcal{S}_F \cup \mathcal{S}_U$. We call $\mathcal{S}^*$ the reference Pareto frontier of the original problem built by a brute-force algorithm that, in our case of n agents with d values in each of their domains and of $n+1$ constraints, has worst-case time complexity $\mathcal{O}((n+1) \cdot d^n)$. Note that some solutions in $\mathcal{S}$ can dominate some solutions in $\mathcal{S}^*$ due to the cut of some physical (hard) constraint in making the factor graph acyclic. To assess the quality of the solutions provided by our approach, we use three different metrics, namely generational distance, hypervolume indicator, and violation degree.

Generational distance. This metric can be applied only when $|\mathcal{S}_F| > 2$. For each point of $\mathcal{S}_F$, the distance to the closest point of $\mathcal{S}^*$ is calculated. The generational distance is the mean of these distances. The main weakness of this metric is that it does not capture the amount of covering of $\mathcal{S}^*$ provided by $\mathcal{S}$.

Hypervolume indicator. This metric tries to overcome the previous drawback by measuring the (normalized) inverse difference between the volume enclosed by $\mathcal{S}^*$ and that enclosed by $\mathcal{S}_F$ in the (hyper)space of objective functions. It is a value in $[0, 1]$, and the closest to 1 the better the approximation of $\mathcal{S}^*$ provided by $\mathcal{S}_F$. Also hypervolume can be calculated only when $|\mathcal{S}_F| > 2$.

Violation degree. This metric is calculated on $\mathcal{S}$ without referring to $\mathcal{S}^*$. It measures the relative amount of violation for a constraint $F_i(\mathbf{x}_i)$ that has been affected by a cut in the original cyclic factor graph. The maximum violation for F_i is $c_i^* = \max_{\mathbf{x}_i} F_i(\mathbf{x}_i)$, while the actual violation of F_i caused by (the assignments in) a solution $\mathcal{S}$ is c_i. The violation degree for F_i is then $v_i = \frac{c_i}{c_i^*}$. If multiple constraints are cut, we calculate the average of their v_i.

The first two metrics evaluate the convergence and diversity of the solutions with respect to the Pareto frontier [15], while the third one accounts for the feasibility of the solutions.

The MO-Max-Sum algorithm has been implemented in Java with the initial problems provided as a XML file. For experiments reported here, we use a computer equipped with an Intel Core i5-2410M 2.30 GHz CPU, 6 GB RAM, and Windows 7 64 bit as operating system.

		Average computation time [s]	
# agents	$\lvert D_i \rvert$	MO-Max-Sum	Brute-force
5	5	0.19	0.58
6	6	0.25	5.87
6	8	0.35	33.64
6	10	0.45	136.80
7	6	0.45	294.69

Table 4: Computation times for acyclic graphs

5.2 Results

We first discuss some results obtained with acyclic factor graphs for which MO-Max-Sum finds the optimal solution (all the points on the Pareto frontier). For each combination of number of agent (5, 6, or 7) and of number of elements in their domains (from 5 to 10), we generate 7 random water resources systems with associated acyclic factor graphs and tested them in 3 flow scenarios (i.e., high, medium, and low flow conditions). In all the cases, we verify that $\mathcal{S} = \mathcal{S}^*$, as expected. In this case, generational distance is 0, hypervolume is 1, and violation degree is meaningless. The computation times for some of the combinations are reported in Table 4, which shows that, for acyclic graphs, MO-Max-Sum scales much better than the brute-force algorithm, according to the worst-case theoretical analysis of their computational complexity.

We now consider more interesting problems with associated cyclic factor graphs. In these cases, the proposed MO-Max-Sum algorithm is not guaranteed to find all the points on the Pareto frontier $\mathcal{S}^*$. For example, Figure 4 (top) shows the inverse of the generational distance metric averaged over 7 randomly generated water systems and 3 flow scenarios (total 21 scenarios) for the case of 8 agents, all with domains of 7 elements. High values mean good solutions. The hypervolume metric for the same setting is reported in Figure 4 (middle). Both figures show the effects of the weighting techniques: RU seems to perform better, very close to U and CL, in producing good solutions, namely in cutting edges of the original factor graph that have little influence on the quality of the obtained Pareto frontier $\mathcal{S}$. Note that sometimes the hypervolume metric is null. This is because some solutions in $\mathcal{S}$ are degenerate, namely they are located on the axes of the hyperspace of objective functions. When $|\mathcal{S}_F| \leq 2$, the two above metrics cannot be calculated, so in these cases the violation degree can provide some information (Figure 4 (bottom) shows results for 16 out of 21 scenarios in the same setting previously considered). The average violation is always smaller than 15%. The variability of the violation degree for different weighting techniques depends on the edges they cut. For example, for scenario 8 of Figure 4 (bottom), solutions on $\mathcal{S}$ do not violate any constraint corresponding to edges cut in making the factor graph of the scenario acyclic. From this perspective, the best weighting technique appears to be RN. These results are confirmed also by the values of the maximum violation degree (not shown here).

We now analyze in more detail when $\mathcal{S}$ does not contain any solution that dominates a solution in $\mathcal{S}^*$. This case intuitively corresponds to a well-behaved cut of edges during the construction of the MST. In the case of 8 agents with 7 values in each of their domains, we generate 40 random water

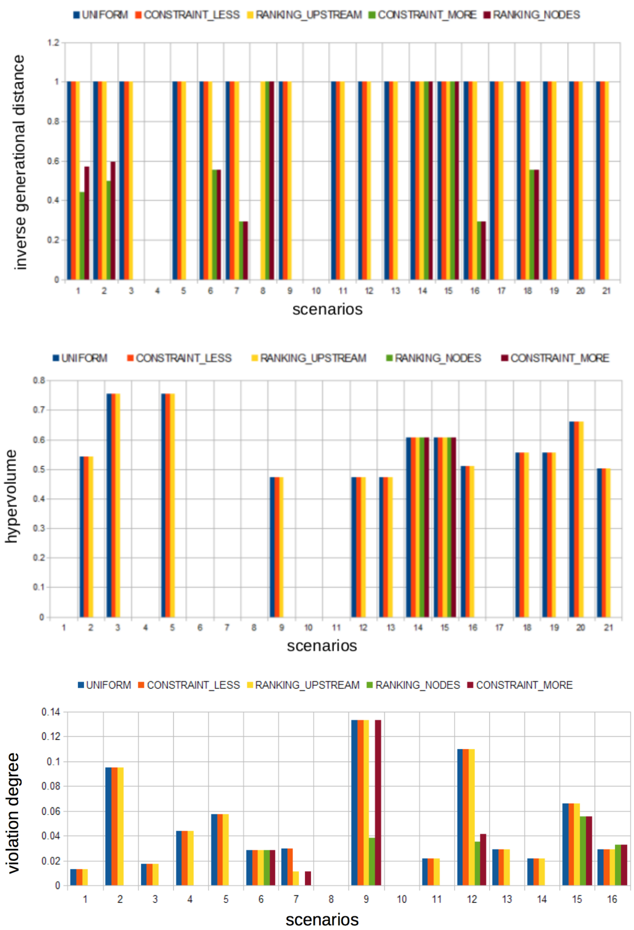

Figure 4: Average inverse generational distance (top), average hypervolume (middle), and average violation degree (bottom)

resources systems. For 12 of them, there is at least a weighting technique for which MO-Max-Sum produces solutions $\mathcal{S}$ completely dominated by $\mathcal{S}^*$. Globally, this happens in 26 cases out of $200 = 40 \times 5$, where 5 is the number of weighting techniques we consider. In all these cases, $\mathcal{S} = \mathcal{S}_F$, namely all the solutions are feasible.

Finally, we evaluate the computation effort of the MO-Max-Sum algorithm in the case of cyclic factor graphs. We note that the time required by MST algorithm is negligible with respect to the time required by the rest of the MO-Max-Sum algorithm. In a distributed algorithm like MO-Max-Sum, the number of messages that agents exchange is critical. We first consider domains with 8 values and, for a given number of agents, we generate 9 water systems that are tested in the 3 flow scenarios. Table 5 (top) reports the average results that show how the weighting techniques RN and CM can better limit the amount of messages when the number of agents increases. When considering 5 agents and varying the size of their domains, we obtain the results of Table 5 (bottom) from which no clearly dominating weight technique can be identified.

Using the same settings as before, we also evaluate the computation times when varying the number of agents (Table 6 (top)) and the sizes of their domains (Table 6 (bottom)). The first table suggests that U and RN weighting techniques can provide some advantages when the number of agents grows and the second table supports these findings with respect to the domains' size of the agents' variables.

	Average number of messages				
# agents	U	RU	RN	CL	CM
5	192	188	189	187	182
6	273	253	194	252	191
7	343	333	292	326	280
8	393	394	402	394	403
9	509	542	408	474	397
10	639	725	335	749	326

	Average number of messages				
$\lvert D_i \rvert$	U	RU	RN	CL	CM
5	182	181	173	186	175
10	198	198	187	200	185
15	199	195	183	195	183
20	199	199	191	196	185
25	199	188	181	191	183
30	194	196	191	195	193

Table 5: Average number of messages vs. number of agents (top) and size of domains (bottom)

	Average computation time [s]				
# agents	U	RU	RN	CL	CM
5	0.33	0.51	0.93	1.09	1.20
6	0.42	1.73	1.84	2.42	3.14
7	1.20	1.68	1.98	0.37	1.70
8	0.88	4.96	6.77	6.57	7.09
9	2.14	8.83	6.69	7.67	7.40
10	21.46	27.79	14.21	31.26	26.36

	Average computation time [s]				
$\lvert D_i \rvert$	U	RU	RN	CL	CM
5	0.13	1.30	2.99	0.16	0.54
10	0.36	1.21	1.07	0.32	0.77
15	0.68	1.48	2.02	0.53	0.97
20	2.57	5.77	7.61	9.27	11.13
25	5.51	9.38	12.29	15.92	16.95
30	14.57	28.27	31.71	45.47	41.46

Table 6: Average computation time vs. number of agents (top) and size of domains (bottom)

6. CONCLUSIONS

In this paper we presented an approach to model water resources management problems as Multi-Objective DCOPs and to identify the whole Pareto frontier using the MO-Max-Sum algorithm, which is a variant of the B-MOMS algorithm proposed in literature. An acyclic factor graph that represents the structure of the problem is required by MO-Max-Sum and this graph is calculated by a minimum spanning tree algorithm using different definitions of weights. We experimentally found that, in the field of water resources systems, RN weighting techniques seems to attain a good trade-off between solution quality and computation effort.

Future work will address the further development of the proposed model, mainly in the direction of including strategic behavior of agents, moving toward a game-theoretical model. The application to a real-world case study will finally allow the validation of the method through the interaction with the real stakeholders and decision-makers and the extension to dynamic problems.

Acknowledgments

Authors gratefully thank the contributions of Enrico Bontempi and Stefano Suardi to parts of this study.

REFERENCES

[1] L. An. Modeling human decisions in coupled human and natural systems: review of agent-based models. *Ecological Modelling*, 229:25–36, 2012.

[2] I. Athanasiadis. A review of agent-based systems applied in environmental informatics. In *Proceedings of the MODSIM International Congress on Modelling and Simulation*, pages 1574–1580, 2005.

[3] E. Bonabeau. Agent–based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287, 2002.

[4] F. Bousquet and C. Le Page. Multi–agent simulations and ecosystem management: a review. *Ecological Modelling*, 176(3-4):313–332, 2004.

[5] E. Brill, J. Flach, L. Hopkins, and S. Ranjithan. MGA: A decision support system for complex, incompletely defined problems. *IEEE Transactions on Systems, Man and Cybernetics*, 20(4):745–757, 1990.

[6] F. Delle Fave, R. Stranders, A. Rogers, and N. Jennings. Bounded decentralised coordination over multiple objectives. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 371–378, 2011.

[7] J. Galán, A. López-Paredes, and R. Del Olmo. An agent-based model for domestic water management in valladolid metropolitan area. *Water Resources Research*, 45(5), 2009.

[8] R. Gallager, P. Humblet, and P. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, 1983.

[9] M. Giuliani and A. Castelletti. Assessing the value of cooperation and information exchange in large water resources systems by agent-based optimization. *Water Resources Research*, 49(7):3912–3926, 2013.

[10] M. Giuliani, A. Castelletti, F. Amigoni, and X. Cai. Multiagent systems and distributed constraint reasoning for regulatory mechanism design in water management. *Journal of Water Resources Planning and Management*, 2014.

[11] I. Huskova and J. Harou. An agent model to simulate water markets. In *Proceedings of the International Congress on Environmental Modeling and Software (iEMSs 2012)*, 2012.

[12] L. Kanta and E. Zechman. Complex Adaptive Systems Framework to Assess Supply-Side and Demand-Side Management for Urban Water Resources. *Journal of Water Resources Planning and Management*, 140(1):75–85, 2014.

[13] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[14] Q. Le, R. Seidl, and R. Scholz. Feedback loops and types of adaptation in the modelling of land–use decisions in an agent–based simulation. *Environmental Modelling & Software*, 27-28:83–96, 2012.

[15] H. Maier, Z. Kapelan, J. Kasprzyk, J. Kollat, L. Matott, M. Cunha, G. Dandy, M. Gibbs, E. Keedwell, A. Marchi, A. Ostfeld, D. Savic, D. Solomatine, J. Vrugt, A. Zecchin, B. Minsker, E. Barbour, G. Kuczera, F. Pasha, A. Castelletti, M. Giuliani, and P. Reed. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environmental Modelling and Software*, 62:271–299, 2014.

[16] T. Matsui, M. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo. Distributed search method with bounded cost vectors on multiple objectives DCOPs. In *Proceedings of the 15th International Conference on Principles and Practice of Multi-Agent Systems*, pages 137–152, 2012.

[17] P. Modi, W. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.

[18] T. Ng, J. Eheart, X. Cai, and J. Braden. An agent-based model of farmer decision-making and water quality impacts at the watershed scale under markets for carbon allowances and a second-generation biofuel crop. *Water Resources Research*, 47(9), 2011.

[19] N. Nguyen, J. Shortle, P. Reed, and T. Nguyen. Water quality trading with asymmetric information, uncertainty and transaction costs: A stochastic agent-based simulation. *Resource and Energy Economics*, 35(1):60–90, 2013.

[20] A. Rogers, A. Farinelli, R. Stranders, and N. Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[21] P. Schreinemachers and T. Berger. An agent–based simulation model of human–environment interactions in agricultural systems. *Environmental Modelling & Software*, 26(7):845–859, 2011.

[22] M. Shafiee and E. Zechman. An agent-based modeling framework for sociotechnical simulation of water distribution contamination events. *Journal of hydroinformatics*, 15(3):862–880, 2013.

[23] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Princeton University Press, 2009.

[24] G. Weiss. *Multiagent Systems*. MIT Press, 2013.

[25] D. Whittington, X. Wu, and C. Sadoff. Water resources management in the Nile basin: the economic value of cooperation. *Water Policy*, 7:227–252, 2005.

[26] Y. Yang, X. Cai, and D. Stipanoviȅ. A decentralized optimization algorithm for multiagent system-based watershed management. *Water Resources Research*, 45(8):1–18, 2009.

[27] Y. Yang, J. Zhao, and X. Cai. Decentralized optimization method for water allocation management in the yellow river basin. *Journal of Water Resources Planning and Management*, 138(4):313–325, 2012.