Exploiting Answer Set Programming for Handling Information Diffusion in a Multi-Social-Network Scenario

Giuseppe Marra[1], Francesco Ricca[2], Giorgio Terracina[2], and Domenico Ursino[1,⋆]

[1] DIIES, University Mediterranea of Reggio Calabria, Via Graziella, Località Feo di Vito, 89122 Reggio Calabria, Italy
[2] Dipartimento di Matematica, University of Calabria, Via Pietro Bucci, 89136 Rende (CS), Italy

Abstract. In this paper we apply Answer Set Programming for analyzing properties of social networks, and we consider Information Diffusion in Social Network Analysis. This problem has been deeply investigated for single social networks, but we focus on a new setting where many social networks coexist and are strictly connected to each other, thanks to those users who join more social networks. We present some experiments allowing us to conclude that the way of spreading information in a Multi-Social-Network scenario is completely different from that of a Single-Social-Network context.

1 Introduction

Answer Set Programming (ASP)[2,14,18,30,31] is a powerful programming paradigm for knowledge representation and declarative problem-solving. The idea of ASP is to represent a given computational problem by a logic program such that its answer sets correspond to solutions, and then, use an answer set solver to find such solutions. The high knowledge-modeling power [2,14] of ASP and the availability of efficient ASP systems [11], make ASP a suitable choice for implementing applications where there is the need of representing and manipulating complex knowledge. Nowadays, ASP counts applications in several fields, ranging from Artificial Intelligence [1,3,4,17,32] to Knowledge Management [2,5], Information Integration [8,7,28,29], and it was also exploited in industrial applications [21,22]. In this paper we apply ASP in a further field, namely Social Network Analysis [12,16]. In particular, we focus on one of the most relevant problems in this field, called Information Diffusion [13,19,20,24,25,26].

Information Diffusion problem has been investigated in the past for a Single-Social-Network context. According to [24], this problem can be divided into three main issues, namely: *(i) modeling diffusion process*, which implies to determine how (i.e., through which paths) information is spread, *(ii) detecting influential*

⋆ This work was partially supported by Aubay Italia S.p.A. and by the project BA2Kno (Business Analytics to Know) funded by MIUR.

E. Fermé and J. Leite (Eds.): JELIA 2014, LNAI 8761, pp. 618–627, 2014.

nodes, which requires to identify those nodes of the network that play important roles in the spreading process, and *(iii) analyzing the most diffused topics*, which concerns the detection of the most popular pieces of information within the network and those appearing the most relevant for a given node.

As for the *diffusion process modeling*, a basic predictive model is the *Linear Threshold* (LT) one [20]; it assumes the existence of a static graph (representing the social network) through which the diffusion process proceeds. LT requires the definition of an influence degree on each edge and of a threshold on each node. The diffusion process iteratively proceeds by starting from a set of initially *activated* nodes. Inactive nodes are activated only if the sum of the degrees of the edges directly connected to active nodes is higher than to the corresponding node threshold. An alternative predictive model is the *Independent Cascade* (IC) one [19]. In this model only edge weights play a role in the Information Diffusion process. Indeed, once activated, a node has a unique chance to activate an inactive neighbor node; this chance is directly proportional to the weight of the edge connecting them. More recent models [23,33] improve these seminal ones, allowing, for instance, the relaxation of the synchronicity assumption, previously mandatory.

As for the *detection of influential nodes*, in the past, a variety of approaches facing it in a single social network have been proposed. For instance, Kempe et al. [25,26] propose an approach that exploits both LT and IC to face the *influence maximization* problem. This problem was first introduced in [13]. Given a parameter k, it aims at finding the k maximally influential nodes (i.e., the k best early adopters). Indeed, thanks to a correct choice of them, it is possible to trigger a large Influence Cascade within the network. Furthermore, found solutions can be used to extract some general features characterizing them (i.e., a sort of their "identikit"). We call this side-problem *influential node characterization* and its extension (and next solution) from a Single-Social-Network Context to a MSNS is one of the main contributions of this paper.

Information Diffusion has been largely investigated in the past on single social networks. However, the current scenario is Multi-Social-Network [6,9,10]. Here, many social networks coexist and are strictly connected to each other, thanks to those users who join more social networks, acting as bridges among them. But, what happens to the Information Diffusion problem when passing to this new scenario? New aspects must be taken into account and new considerations are in order. However, to the best of our knowledge, no investigation about this issue has been made in the past. When starting this task, several new questions arise, such as: *(i)* What is the role of bridges for Information Diffusion in a Multi-Social-Network Scenario (MSNS, for short)? *(ii)* Are there other kind of nodes (such as power user or bridge's direct neighbors) that play a key role in Information Diffusion? *(iii)* What is the "identikit" of the most influential nodes? *(iv)* How this identikit varies when the number of social networks of the MSNS increases? In this paper, we exploit ASP to give an answer to these questions and, more in general, to face the Information Diffusion problem in an MSNS.

2 Answer Set Programming

ASP is a declarative programming paradigm based on nonmonotonic reasoning. Its main advantage consists in its declarativity, combined with a relatively high expressive power. In ASP, a *(disjunctive) rule* r has the following form:

$$a_1 \vee \ldots \vee a_n \coloneq b_1, \ldots, b_k, \texttt{not}\ b_{k+1}, \ldots, \texttt{not}\ b_m.$$

where $a_1, \ldots, a_n, b_1, \ldots, b_m$ are atoms, and $n, k, m \geq 0$. A literal is either an atom a or its negation $\texttt{not}\ \ a$. The disjunction $a_1 \vee \ldots \vee a_n$ is the *head* of r, while the conjunction $b_1, \ldots, b_k, \texttt{not}\ b_{k+1}, \ldots, \texttt{not}\ b_m$ is its *body*. Rules with empty body are called *facts*. Those with empty head are called *strong constraints*. A rule is *safe* if every variable occurs in some positive literal of the body. An ASP program is a set of safe rules. An atom, a literal, a rule, or a program is *ground* if no variables appear in it. Let P be an ASP program. The *Herbrand universe* U_P and the *Herbrand base* B_P of P, are defined as usual. The ground instantiation G_P of P is the set of all the ground instances of rules of P, that can be obtained by substituting variables with constants from U_P. An *interpretation* I for P is a subset I of B_P. A ground literal ℓ (resp. $\texttt{not}\ \ell$) is true w.r.t. I if $\ell \in I$ (resp. $\ell \notin I$), and false (resp. true) otherwise. A ground rule r is *satisfied* by I if at least one atom in the head is true w.r.t. I whenever all literals in the body of r are true w.r.t. I. A model is an interpretation that satisfies all the rules of a program. Given a ground program G_P and an interpretation I, the *reduct* [15] of G_P w.r.t. I is the subset G_P^I of G_P obtained by deleting from G_P the rules in which a body literal is false w.r.t. I. An interpretation I for P is an *answer set* (or stable model [18]) for P if I is a minimal model (under subset inclusion) of G_P^I (i.e., I is a minimal model for the program G_P^I) [15]. Optimal answer sets can be specified by weak constraints. An ASP program with weak constraints is $\Pi =< R, W >$, where R is a program and W is a set of weak constraints. In detail, a *weak constraint* ω is of the form:

$$:\sim b_1, \ldots, b_k, \texttt{not}\ b_{k+1}, \ldots, \texttt{not}\ b_m.[w@l]$$

where w and l are the weight and level of ω. The semantics of Π extends from the basic case defined above, thus we assume that R and W are ground in the following. A constraint ω is violated by an interpretation I if all literals in ω are true w.r.t. I. An *optimal answer set* O for P is an answer set of R that minimizes the sum of the weights of the violated weak constraints in a prioritized way.

A complete description of the ASP language is out of the scope of this paper; we refer the reader to [2] for a textbook on Answer Set Programming and to [27] for a complete description of the language implemented by DLV, the ASP implementation used for our analysis, which also supports aggregate atoms [15] to easily encode aggregate functions as the ones available in SQL.

3 Modeling a Multi-Social-Network Scenario

A Multi-Social-Network Scenario models a context where several social networks coexist and are strictly connected to each other, thanks to those users who join more social networks. Indeed, when a user joins more social networks, her multiple accounts allow these networks to be connected. We call *bridge user* each user joining more social networks, *bridge (node)* each account of such a user and `me` *edge* each edge connecting two bridges.

A Multi-Social-Network Scenario Ψ, consisting of n social networks $\{S_1, S_2, \ldots, S_n\}$, can be modeled by a pair $\langle G, T\rangle$. Here, T is a list $\{t_1, t_2, \ldots, t_p\}$ of topics of interest for the users of Ψ. It is preliminarily obtained by performing the union/reconciliation of the topics related to the social networks of Ψ. G is a graph and can be represented as $G = \langle V, E\rangle$. V is the set of nodes. A node $v_i \in V$ represents a user account in a social network of Ψ. $E = E_f \cup E_m$ is a set of edges. E_f is the set of friendship edges; E_m is the set of `me` edges. An edge $e_j \in E$ is a triplet $\langle v_s, v_t, L_j\rangle$. v_s and v_t are the source and the target nodes of e_j, whereas L_j is a list of p pairs $\langle t_{j_k}, w_{j_k}\rangle$, where t_{j_k} is a topic and w_{j_k} is a real number between 0 and 1 representing the corresponding weight. This weight depends on both t_{j_k} and the ability of the user associated with v_t to propagate, to the user associated with v_s, the information related to t_{j_k}.

4 Formalizing the Information Diffusion Problem in a Multi-Social-Network Scenario

As previously pointed out, to extend the information diffusion problem from a single social network to an MSNS, it is necessary to consider the peculiarities of this scenario. As for the first issue of the Information Diffusion problem (i.e., the *diffusion process model*), we chose to exploit the Linear Threshold model. Our updated version of this model in MSNS works as the traditional one, except for `me` edges. In fact, as said before, a `me` edge links two accounts of the *same* user (i.e. bridge nodes) belonging to different social networks. Thus, it makes no sense to talk of *influence degree* for these edges, since a user cannot influence herself. Actually, we can still define a degree for `me` edges but it depends on the probability of a bridge user to share the content in other social networks joined by her (i.e., to spread the information from a social network to another one). This probability is a function of both the habits of the bridge user and the features of the two social networks she joins. Moreover, as for the activation rule of bridge nodes, the definition of a threshold is misleading. Indeed, given a `me` edge and a bridge, if this last does not activate the corresponding bridge at the moment of its own activation, it's unrealistic that it will do this task in a second time. As a consequence, it is reasonable to adopt an activation policy for `me` edges similar to the one suggested by the Independent Cascade model. This means that, at the time of its activation, given a `me` edge and a bridge, this last has a single chance proportional to the probability defined for the edge, to activate the corresponding bridge. On the basis of this reasoning, our diffusion process model (called MSNS-DP model) is as follows.

MSNS-DP model. Consider an Information Diffusion task in an MSNS and assume that at the j^{th} step some nodes have already been activated. At the $(j+1)^{th}$ step an inactive node n is activated if: *(i)* the sum of the degrees of friendship edges directly connecting n to already active neighbors is higher than the threshold associated with n, and/or *(ii)* a random number uniformly extracted in the interval $[0, 1]$ is lower than the diffusion probability of a `me` edge connecting n to a bridge activated at the j^{th} step.

As for the second issue of the Information Diffusion problem (i.e., *the detection and characterization of influential spreaders*), we start from the influence maximization problem introduced in the Introduction. However, even in this case, some modifications are in order. In fact, when passing from a single social network to an MSNS, it could happen that the optimal solution found by classical approaches maximizes the diffusion in a single network leaving uncovered the remaining ones. In order to take the peculiarities of the MSNS into account, a slightly different definition of the influence maximization problem (called MSNS-IM problem) is required.

MSNS-IM Problem. Given in input:

- A Multi-Social-Network Scenario Ψ, made of n social networks $\{S_1, \ldots, S_n\}$.
- A list D of n elements. The generic element D_h of D consists of a tuple $\langle S_h, p_h, c_h \rangle$. Here, S_h is a social network of Ψ. c_h is the minimum desired coverage for S_h, i.e., the minimum number of nodes of S_h which must be reached by the information to spread throughout Ψ. p_h denotes the priority of S_h, it is an integer from 1 to n, where 1 (resp., n) is the maximum (resp., minimum) priority. The social network with the maximum (resp., minimum) priority will the first (resp., the last) to have its coverage requirements satisfied.
- A list τ of q elements. The generic element $\tau[k]$ of τ is a pair $\langle t_k, \omega_k \rangle$. Here, t_k corresponds to the k^{th} element of the set of topics T of Ψ. ω_k is a real number, belonging to the interval $[0, 1]$ and indicating the weight of t_k in the information to spread throughout Ψ.

The MSNS-IM problem in Ψ requires to find the minimum set of the nodes of Ψ allowing the maximization of the coverage of the social networks of Ψ, taking into account the minimum required network coverage, the network priorities (as expressed in D), and the topics characterizing the information to spread (as expressed in τ). Observe that this version of the problem is quite different from the one specified for single social networks in the past. Indeed, it does not fix the parameter k but asks to find the minimum set of nodes (i.e., minimizing k) that are able to trigger a diffusion process that guarantees, at least, the coverage requirements represented in D. In this way, the optimization task is transferred to the number of earlier starters, whereas the maximization of the overall coverage is not considered, since it makes no sense in an MSNS. Clearly, the solution of MSNS-IM problem, along with a next study of returned nodes, leads to the detection and characterization of influential spreaders and, therefore, to face the second issue of the Information Diffusion problem. Finally, in our definition, topics (i.e., the third issue of the Information Diffusion problem - see the Introduction) can be handled by means of the list τ given in input. In this way, once the topics of interest of each node have been determined, it is possible to state how much a node is important in the Information Diffusion process into consideration.

5 Handling Information Diffusion in a MSNS with ASP

The MSNS-IM problem described in the previous section is extremely complex. The adoption of ASP has been a strategic choice to allow an easy modeling and a fast set-up of the approach implementation. Interestingly, the elegant modeling of the problem in ASP is associated with acceptable performances of the implementation.

First, let us define the input format of the problem. Let `starting_node(V)` be the set of nodes from which initially *activated* nodes must be chosen. Let `edge(V1,V2,K)` be the relation containing the edges from `V1` to `V2`, where `K` specifies the edge kind (i.e., `me` or `friendship`). Let `edge_topic(V1,V2,T,W)` be the set of topics/weights associated with the edge from v_1 to v_2. Let `node(V,Sn)` represent the set of nodes in the social network S_n. Finally, let `D(Sh,Ph,Ch)` identify the desiderata for coverage and priority and `tau(T,W)` the set of topics, with the corresponding weights, of the information to spread.

The logic program designed to solve our problem is as follows

```
 1. in(V) v out(V) :- starting_node(V).
 2. :- D(Sh,Ph,Ch), #count{V:active(V),node(V,Sh)}<=Ch.
 3. active(V) :- in(V).
 4. active(V) :- active(V1),edge(V,V1,me).
 5. active(V) :- node(V,Sn), #sum{W: edge(V,V1,friendship), active(V1),
             edge_topic(V,V1,T,W), tau(T,Wb) }>=Tw
 6. :~ in(V). [1@4]
 7. :~ node(V,Sn), not active(V). [1@3]
 8. :~ D(S1,P1,C1), D(S2,P2,C2), P1<P2,
             nactive(S1,N1), nactive(S2,N2), N1<N2. [1@2]
 9. nactive(Sn,N) :- node(V,Sn), #count{W: active(W), node(W,Sn)}=N.
10. :~ tau(Ta,Wa), tau(Tb,Wb), active(V1), active(V2), Wa>Wb,
             edge_topic(V1,V2,Ta,W1), edge_topic(V1,V2,Tb,W2), W1<W2. [1@1]
```

where, rule 1. guesses a subset of starting nodes sufficient for the optimization purposes. To discard non admissible solutions, constraint 2. is exploited. In order to compute the nodes activated by the current choice, rules 3. to 5. are applied. Rules 3. and 4. state that a node is active if either it is a starting one, or it reaches an active node through a `me` edge. In rule 5., `Tw` is a fixed threshold indicating the minimum weight that must be totalized through the topics of the edges connected to `V` to activate it. In our experimental campaign we performed some simplifications about this activation policy. In particular, we assumed that all the topics of Ψ have the same weight.It follows that all the friendship edges in Ψ have the same weight (we assign a weight equal to 1 to them). We also assumed that `me` edges always propagate the information to spread. This means assigning a weight equal to 1 to all `me` edges. Finally, we assumed also that a node is activated when at least two edges, outgoing from it, are pointing to already activated nodes. This corresponds to set the threshold `Tw` to 2. Under these assumptions, rules 3.-5. can be simplified.

Returning to the examination of our approach, we point out that the optimization step consisting of the choice of the best models among the consistent ones, is carried out by a number of weak constraints. Specifically, the weak constraint 6. imposes that the number of nodes *in* the consistent solutions must be minimum. The weak constraint 7. imposes the minimization of non-active nodes, whereas the weak constraint 8. states that the best solution must be such that the order of Social Networks in terms of activated nodes must follow what specified in the desiderata or, at least, the number of violations of this order must be minimized. Rule 9. is an auxiliary one, counting the number of active nodes for each social network. Analogously, weak constraint 10. minimizes the number of selected arcs whose list of topics does not comply with the topic classification specified in `tau`. Observe that all the weak constraints have the same weight, but different priorities. This guarantees that, for instance, the minimum sets of nodes providing consistent solutions are identified first, and, among them, the ones minimizing non-active nodes are selected.

6 Experimental Campaign

To test our Information Diffusion approach we performed an experimental campaign on an MSNS consisting of four social networks, namely LiveJournal, Flickr, Twitter and YouTube. We chose these networks because they are the ones allowing an easier access to their own data. Our MSNS has 93177 nodes and 146957 edges. The dataset can be downloaded from: `www.ursino.unirc.it/DiffusionJELIA.html`. The password the Reader must specify is "85749236".

We performed a large number of runs of our ASP program using DLV [27]. In these runs we considered many configurations of the starting nodes. They differed in the number of nodes (ranging from 25 to 100 with a step of 25), the percentage of bridges (ranging from 0 to 100 with a step of 10), and the number of the social networks to cover (ranging from 2 to 4). To reduce the influence of possible outliers, for each configuration we considered four different sets of starting nodes randomly constructed by following the guidelines discussed in Section 4. For each set of starting nodes we considered 10 different network coverage requirements (ranging from 10% to 100% of each social network with a step of 10%). The whole number of runs we have performed was 5280. Due to space limitations, in the following we report only some of obtained results.

As a first experiment we measured the average percentage of bridges in the optimal solutions. For this purpose, we computed the variation of the average percentage of bridges present in the optimal solutions against the variation of the average percentage of bridges present in the sets of starting nodes. Obtained results are shown in Figure 1. Observe that the percentage of bridges in the optimal solutions is generally higher, or much higher, than the percentage of bridges in the sets of starting nodes. This information is precious for drawing an identikit of the most influential nodes for Information Diffusion in an MSNS. Indeed, it suggests that bridges certainly play a key role in Information Diffusion in an MSNS. Therefore, a first feature of influential nodes is that they are generally bridges. Observe that, while it is straightforward that bridges are important for spreading information from a social network to another of an MSNS, it is not so

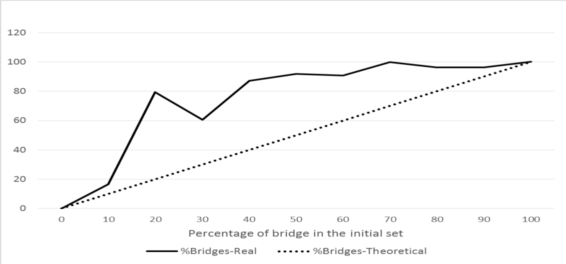

Fig. 1. Average percentage of bridges in the optimal solutions

obvious that starting nodes are generally bridges. Indeed, in principle, it could happen that starting nodes are non-bridges, and bridges are reached only in a second time. The fact that starting nodes are generally bridges is an important result of our paper and indicates that bridges allow the minimization of the set of nodes necessary for spreading information in an MSNS.

As a second experiment we analyzed which kind of nodes generally compose the optimal solutions. For this purpose, we computed the following statistics (we first report the parameters and, then, in parentheses, the obtained value): *(i)* average percentage of bridges (87%); *(ii)* average percentage of the direct neighbors of bridges (13%); *(iii)* average percentage of power users (83%); *(iv)* average percentage of the direct neighbors of power users (7%); *(v)* average percentage of nodes being both bridges and power users (77%); *(vi)* average percentage of nodes being bridges or power users (93%); *(vii)* average percentage of nodes being bridges but not power users (6%); *(viii)* average percentage of nodes being power users but not bridges (7%); *(ix)* average percentage of nodes being neither bridges nor power users (10%); *(x)* average Jaccard coefficient[1] of bridges and power users (82%). From the analysis of these values we can observe that 100% of the nodes in the optimal solutions are either bridges or direct neighbors of bridges. Analogously, 90% of the nodes in the optimal solutions are either power users or direct neighbors of power users. Furthermore, the majority of the bridges involved in the optimal solutions are power users, and vice versa. Finally, only a little fraction of the nodes present in the optimal solutions are neither bridges nor power users. These results allow us to conclude that almost all the bridges in the optimal solutions are power users. It tells also that if an influential node is not a bridge, it is surely a direct neighbor of a bridge. We think that the capability of our approach of finding solutions with a low number of node, as emerged in the first test, is due to the double nature of influential nodes: as bridges they can start the Information Diffusion process among the social networks of our MSNS; as power users they can favor the in-depth diffusion of the same information.

[1] We recall that the Jaccard Coefficient $J(A, B)$ between two sets A and B is defined as $J(A, B) = \frac{A \cap B}{A \cup B}$.

References

1. Balduccini, M., Gelfond, M., Watson, R., Nogueira, M.: The USA-Advisor: A Case Study in Answer Set Planning. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 439–442. Springer, Heidelberg (2001)
2. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
3. Baral, C., Gelfond, M.: Reasoning Agents in Dynamic Domains. In: Logic-Based Artificial Intelligence, pp. 257–279. Kluwer Academic Publishers (2000)
4. Baral, C., Uyan, C.: Declarative Specification and Solution of Combinatorial Auctions Using Logic Programming. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 186–199. Springer, Heidelberg (2001)
5. Bardadym, V.A.: Computer-Aided School and University Timetabling: The New Wave. In: Burke, E., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 22–45. Springer, Heidelberg (1996)
6. Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., Pedreschi, D.: The pursuit of hubbiness: Analysis of hubs in large multidimensional networks. Journal of Computational Science 2(3), 223–237 (2011)
7. Bertossi, L., Hunter, A., Schaub, T. (eds.): Inconsistency Tolerance. LNCS, vol. 3300. Springer, Heidelberg (2005)
8. Bravo, L., Bertossi, L.: Logic programming for consistently querying data integration systems. In: Proc. of the International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, pp. 10–15 (2003)
9. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Discovering Links among Social Networks. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 467–482. Springer, Heidelberg (2012)
10. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Moving from social networks to social internetworking scenarios: The crawling perspective. Information Sciences 256, 126–137 (2014)
11. Calimeri, F., Ianni, G., Ricca, F.: The third open answer set programming competition. TPLP 14(1), 117–135 (2014)
12. Carrington, P., Scott, J., Wasserman, S.: Models and Methods in Social Network Analysis. Cambridge University Press (2005)
13. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), pp. 57–66. ACM, San Francisco (2001)
14. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive Datalog. ACM Transactions on Database Systems 22(3), 364–418 (1997)
15. Faber, W., Leone, N., Pfeifer, G.: Recursive aggregates in disjunctive logic programs: Semantics and complexity. In: Alferes, J.J., Leite, J. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 200–212. Springer, Heidelberg (2004)
16. Freeman, L.: The Development of Social Network Analysis. Empirical Press (2006)
17. Garro, A., Palopoli, L., Ricca, F.: Exploiting agents in e-learning and skills management context. AI Communications – The European Journal on Artificial Intelligence 19(2), 137–154 (2006)
18. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing 9, 365–385 (1991)
19. Goldenberg, J., Libai, E., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12(3), 211–223 (2001)

20. Granovetter, M.: Threshold models of collective behavior. American Journal of Sociology 83(6), 1127–1138 (1978)
21. Grasso, G., Iiritano, S., Leone, N., Lio, V., Ricca, F., Scalise, F.: An asp-based system for team-building in the gioia-tauro seaport. In: Carro, M., Peña, R. (eds.) PADL 2010. LNCS, vol. 5937, pp. 40–42. Springer, Heidelberg (2010)
22. Grasso, G., Iiritano, S., Leone, N., Ricca, F.: Some DLV Applications for Knowledge Management. In: Erdem, E., Lin, F., Schaub, T. (eds.) LPNMR 2009. LNCS, vol. 5753, pp. 591–597. Springer, Heidelberg (2009)
23. Guille, A., Hacid, H.: A predictive model for the temporal dynamics of information diffusion in online social networks. In: Proc. of the International World Wide Web Conference (WWW 2012) - Companion Volume, pp. 1145–1152. ACM, Lyon (2012)
24. Guille, A., Hacid, H., Favre, C., Zighed, D.: Information Diffusion in Online Social Networks: A Survey. SIGMOD Record 42(2), 17–28 (2013)
25. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), pp. 137–146. ACM, Washington, DC (2003)
26. Kempe, D., Kleinberg, J., Tardos, É.: Influential Nodes in a Diffusion Model for Social Networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
27. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. ACM Transactions on Computational Logic 7(3), 499–562 (2006)
28. Leone, N., Gottlob, G., Rosati, R., Eiter, T., Faber, W., Fink, M., Greco, G., Ianni, G., Kałka, E., Lembo, D., Lenzerini, M., Lio, V., Nowicki, B., Ruzzi, M., Staniszkis, W., Terracina, G.: The INFOMIX System for Advanced Integration of Incomplete and Inconsistent Data. In: Proceedings of the 24th ACM SIGMOD International Conference on Management of Data (SIGMOD 2005), pp. 915–917. ACM Press, Baltimore (2005)
29. Manna, M., Ricca, F., Terracina, G.: Consistent query answering via ASP from different perspectives: Theory and practice. Theory and Practice of Logic Programming 13(2), 277–252 (2013)
30. Marek, V.W., Truszczyński, M.: Stable models and an alternative logic programming paradigm. CoRR cs.LO/9809032 (1998)
31. Niemelä, I.: Logic Programs with Stable Model Semantics as a Constraint Programming Paradigm. In: Niemelä, I., Schaub, T. (eds.) Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning, Trento, Italy, pp. 72–79 (May/June 1998)
32. Nogueira, M., Balduccini, M., Gelfond, M., Watson, R., Barry, M.: An A-Prolog Decision Support System for the Space Shuttle. In: Ramakrishnan, I.V. (ed.) PADL 2001. LNCS, vol. 1990, pp. 169–183. Springer, Heidelberg (2001)
33. Saito, K., Ohara, K., Yamagishi, Y., Kimura, M., Motoda, H.: Learning diffusion probability based on node attributes in social networks. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 153–162. Springer, Heidelberg (2011)