

**Discovering Chance Scenarios
using Small-World KeyGraphs
and Evolutionary Computation**

**Xavier Llorà, Naohiro Matsumura,
David E. Goldberg, Yukio Ohsawa,
Kei Ohnishi, and Antonio Gonzales**

IlliGAL Report No. 2004026
May, 2004

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-2346
Fax: (217) 244-5705

Discovering Chance Scenarios using Small-World KeyGraphs and Evolutionary Computation

Xavier Llorà¹, Naohiro Matsumura², David E. Goldberg¹,
Yukio Ohsawa³, Kei Ohnishi¹, and Antonio Gonzales¹

¹Illinois Genetic Algorithms Laboratory (IlliGAL),
National Center for Supercomputing Applications,
University of Illinois at Urbana-Champaign, Urbana, IL, 61801
{xllora,deg,kei}@illigal.ge.uiuc.edu

²Graduate School of Economics, Osaka University, Osaka, Japan, 560-0043
matumura@econ.osaka-u.ac.jp

³Chance Discovery Consortium, University of Tsukuba &
University of Tokyo, Tokyo, Japan, 112-0012
osawa@gssm.otsuka.tsukuba.ac.jp

Abstract

A successful process of chance discovery using the visual maps proposed by KeyGraphs requires the usage of graphs with an appropriate degree of complexity. Complex KeyGraphs often prevent users from discovering chances because of the difficulties of interpretation. On the other hand, overly simplistic KeyGraphs seldom includes a chance because of the sparseness of information. In a useful KeyGraphs the concept clusters should be easy to find, the clusters should be easy to understand, and the relations among them should be easy to comprehend and help in the process of chance identification. This paper systematize the process of KeyGraph exploration by means of evolutionary computation, as well as structural graph properties—such as small-world topologies. The proposed techniques are successfully applied to create useful KeyGraphs for chance discovery from several documents.

1 Introduction

The structural complexity of KeyGraphs greatly affects the success of discovering a chance. Complex KeyGraphs often prevent the users from discovering chances because of the difficulties of interpretation. Such difficulties lead users to spend most of their time trying to grasp the relevant concepts present on the KeyGraph instead of focusing on reasoning about relevant chances. On the other hand, an overly simplistic KeyGraph seldom includes a chance because of the sparseness of information.

Building a useful KeyGraph from a given document D greatly depends on the parameters involved in the KeyGraph algorithm (Ohsawa, Benson, & Yachida, 1998; Ohsawa & McBurney, 2003)—still remaining a kind of art. The work presented in this paper focus on systematizing the process of building useful KeyGraph for chance discovery. That is, KeyGraphs where (1)

the clusters are easy to find, (2) the clusters are easy to understand, and (3) the relations among clusters are easily to comprehend and help in the process of chance identification. In order to achieve this things we introduce two evolutionarily-driven KeyGraph exploration—methods for parameter tuning. The first one involves using the users’ judgment for the parameter tuning using interactive genetic algorithms (Takagi, 2001). The second method explores the set of possible KeyGraphs searching for graphs that present a small-world topology (Watts & Strogatz, 1998; Walsh, 1999) using a genetic algorithm (Goldberg, 1989; Goldberg, 2002). Such a topology favors the three success factors to KeyGraph usage mentioned above.

The rest of the paper is structured as follows. Section 2 presents an overview of KeyGraphs for chance discovery. The first method for KeyGraph exploration based on interactive evolutionary computation is introduced in section 3. Section 4 presents the small world concept and introduces how a genetic algorithm may be used to search for small world KeyGraphs. Finally, section 5 discusses the results achieved using such methods and presents final remarks.

2 KeyGraph as a Tool for Chance Discovery

KeyGraphs (Ohsawa, Benson, & Yachida, 1998; Ohsawa & McBurney, 2003) can be applied to documents to obtains a visual map of its contents. The work presented in this paper has used KeyGraphs to analyze on-line web documents and conversation logs of a group collaborating in a creativity session in the DISCUS (*Distributed Innovation and Scalable Collaboration in Uncertain Settings*) project (Goldberg, Welge, & Llorà, 2003; Llorà, Ohnishi, Chen, Goldberg, & Welge, 2004). DISCUS can apply such method, among others, to archived documents, message board messages, or chat room logs, always targeting the discovery of relevant chances on the processed scenarios.

Here we assume that a document D is composed of sentences and each sentence is composed of words. The main steps of the KeyGraph algorithm—for a detailed description please see Ohsawa & McBurney (2003) (Ohsawa & McBurney, 2003)—can be outlined as follows.

Document preprocessing. Consists of two tasks: (1) *document compactation* and (2) *phrase construction*. The first one consists on stop-word removal and word stemming using the Porter algorithm (Porter, 1980). The second tasks takes a subset of ℓ_{phrase} words and all the possible combinations out of those words are constructed, retaining only the high frequency ones appearing in the document. Thus, the document D is reduced to a document D' which contains unique terms w_1, w_2, \dots, w_ℓ , where w_i refers to either a word or a phrase.

Extracting high-frequency terms. Terms in D' are sorted by their fitness. N_{hf} denotes the set of the top n_{nodes} high-frequency terms, being represented as nodes in a graph G .

Extracting links. Links represent *co-occurrence-term-pairs* that often occur in the same sentence. A measure for co-occurrence of terms w_i and w_j is defined as

$$assoc(w_i, w_j) = \sum_{s \in D'} \min(|w_i|_s, |w_j|_s), \quad (1)$$

where w_i and w_j are element of the N_{hf} , and $|w_i|_s$ the number of times a term w_i occurs in a sentence s . The *assoc* values are computed for all term in N_{hf} . The term-pairs are sorted by their *assoc* values and the top n_{links} term-pairs are represented by edges in G .

Extracting key terms. Key terms are terms that connect clusters of high-frequency terms together. To measure the tightness with which a term w connects a cluster, the following

function is defined:

$$key(w) = 1 - \prod_{g \subset G} \left[1 - \frac{based(w, g)}{neighbors(g)} \right] \quad (2)$$

where g is a cluster, and

$$based(w, g) = \sum_{s \in D'} |w|_s |g - w|_s, \quad (3)$$

$$neighbors(w) = \sum_{s \in D'} \sum_{w \in s} |w|_s |g - w|_s, \quad (4)$$

where $|g - w|_s = |g|_s - |w|_s$ if $w \in g$, and $|g|_s$ otherwise, being $|g|_s$ the number of times a cluster g occurs in a sentence s . The *key* values are computed for all the terms in D' and sorted accordingly. The top n_{key} terms—the K_{hk} set—are added as nodes to G if they were not present previously.

Extracting key links. For each high-frequency term $w_i \in N_{hf}$, and each key term $w_j \in K_{hk}$, $assoc(w_i, w_j)$ is calculated. Links touching w_j are sorted by their *assoc* values for each key term $w_j \in K_{hk}$. A link with highest *assoc* values connecting w_j to two or more clusters is chosen as a key link. Key links are represented by edges—if they are not already present—in G .

Extracting Keywords. Nodes in G are sorted by the sum of *assoc* values associated with the key links touching them. Terms represented by nodes of higher values of these sums than a certain threshold are extracted as keywords for the document D .

Twelve different parameters control the process of building a KeyGraph. Parameters such as ℓ_{phrase} , n_{nodes} , and n_{links} , introduced above, control the final overall shape and comprehensibility of KeyGraph outputted. A detailed description of the twelve parameters is beyond the scope of this paper, being accurately explained elsewhere (Ohsawa, Benson, & Yachida, 1998; Ohsawa & McBurney, 2003).

3 Interactive KeyGraphs exploration

Genetic algorithms (GAs) are search procedures based on the mechanics of natural selection and genetics (Goldberg, Korb, & Deb, 1989). They combine

- general and independent evaluation of solution quality or merit
- the coding of the set of possible solutions as a set of alternative chromosomes or genotypes
- the selection of better solutions according to merit
- genetic-like variation mechanisms such as crossover and mutation to promote the rapid generation of new, possible better solutions to a user’s problem

Interactive GAs (iGAs) replace the computer computation of the relative fitness of solutions and the selection process by the judgment of a human evaluation. More detailed information about the progress of interactive GAs and interactive evolutionary computation (iEC) are presented in a review by Takagi (Takagi, 2001).

We created a user interface for interactive parameter adjustment. Two KeyGraphs, generated using different parameters settings, were displayed to the user. The user was asked to select

the KeyGraph that better described the text under analysis. The process was repeated until the user was satisfied. Such model is a straightforward implementation of a (1+1) evolutionary strategy (Rechenberg, 1973; Schwefel, 1977; Bäck, 1996), where the user is in charge of the selection stage. The experiments started displaying two KeyGraphs generated out of two random parameter settings. Researchers evolve such KeyGraphs until they found a KeyGraph where (1) the clusters are easy to find, (2) the clusters are easy to understand, and (3) the relations among clusters are easily to comprehend and help in the process of chance identification.

For testing purposes we have used three different documents: (1) the plot summaries of 1990 *Cyrano de Bergerac* movie¹, (2) a description of the Edgar Degas' *Absinthe* painting², and (3) the logs of marketing discussion about cell phone market using the DISCUS project (Llorà, Goldberg, Ohsawa, Ohnishi, Washida, Tamura, & Yoshikawa, 2004). We strongly recommend the readers to read the first two texts before looking at the KeyGraphs displayed in figures 1. Figures 3 and 4 displays the KeyGraph evolved for the marketing research discussion introduced above (Llorà, Goldberg, Ohsawa, Ohnishi, Washida, Tamura, & Yoshikawa, 2004).

Figures 1 and 3(b) compare the KeyGraphs obtained using the default setting of parameters (Ohsawa, Benson, & Yachida, 1998; Ohsawa & McBurney, 2003) with the ones obtained using interactive evolutionary computation. The evolved KeyGraphs present better cluster identification, understandability, and relevant chances are easier to find. The iEC process requires the active involvement and focus of the researchers. Such involvement helps to clarify the scenario for a chance discovery through the interactive analysis of several KeyGraphs. However, the lack of concentration and users' fatigue tend to introduce noise and drift in such evolutionary process. For that reason, we focus our research on introducing automatic ways for such KeyGraph exploration.

4 KeyGraphs, Small World, and KeyGraphs

The interactive approach presented in the previous section involves human judgment in the search process. Such an approach take advantage of the fact that the users have read the text before, guiding the process based on the set of concepts they consider important, leading to meaningful structures like the ones displayed in figures 1 and 3(a). However, such a process is greatly dependent on users focus and fatigue. In order to achieve such automatic tuning of the KeyGraph parameters some decisions need to be made.

The structural complexity of KeyGraphs greatly affects the success of discovering a chance. Complex KeyGraphs often prevent the users from discovering chances because of the difficulties of interpretation. Such difficulties lead users to spend most of their time trying to grasp the relevant concepts present on the KeyGraph instead of reasoning about relevant chances. On the other hand, a overly simplistic KeyGraph seldom includes a chance because of the sparseness of information. Thus, the creation of a KeyGraph favorable for discovering chances.

The interpretation of a KeyGraph mainly depends on the clarity of the cluster identification, as well as the relations among them. The informative value of a KeyGraph also depends on the amount of interpretation and the effort involved in such interpretation. These kind of structural properties are analogous to the small-world properties of a graph G (Watts & Strogatz, 1998; Walsh, 1999). For such reasons, KeyGraphs having a small-world topology may be ideal candidates to minimize the interpretation effort at the same time that they convey enough useful information to discovery chances.

¹<http://www.imdb.com/title/tt0099334/plotsummary>

²<http://www.ibiblio.org/wm/paint/auth/degas/absinthe/>

The notion of small world in a graph G was formalized by Watts & Strogatz (Watts & Strogatz, 1998) in terms of: (1) the characteristic path length and, (2) the clustering coefficient. The path length is the number of edges in the shortest path between two nodes. The characteristic path length L is the path length averaged over all pairs of nodes. The clustering coefficient is a measure of the cliqueness of the local neighborhoods. Given a node with k neighbors, then at most $k(k-1)/2$ edges can exist between them (this occurs if they form a k -clique). The clustering of a node is the fraction of these allowable edges that occur. The clustering coefficient C is the average of the clustering over all the nodes in the graph. Watts & Strogatz defined a small-world graph G as one in which $L > L_{rand}$ and $C \gg C_{rand}$. L_{rand} and C_{rand} are the characteristic path length and clustering coefficient of a random graph with the same number of nodes and edges. Walsh (Walsh, 1999) propose to compare the topology of two graphs the proximity ratio μ . The μ ratio is defined as the ration C/L normalized by C_{rand}/L_{rand} . In graphs with a small-world topology, the proximity ration $\mu \gg 1$. By comparison, the proximity ration μ is unity in random graphs, and small in regular graphs such as lattices.

The μ ratio was used as the *fitness* function on a genetic algorithm (Goldberg, 1989; Goldberg, 2002). The genetic algorithm used for such purpose used tournament selection ($s = 2$), a one-point crossover operator, and value perturbation mutation operators. The individuals encode the twelve different parameters required by the KeyGraph algorithm. The runs involved 250 individuals and last for 150 generations. At the end of the run, the best individual—the one with higher μ ratio—was selected and the candidate KeyGraph built. We left as a future work to explore the usage of niching (Goldberg, 1989) techniques to maintain different parameter configurations that lead to the same μ ratios.

Figures 2 and 4 present the KeyGraphs obtained using the genetic algorithm with the small-world-based fitness function. The evolved KeyGraphs presents a better μ ratio than the ones obtained using the default configuration and the interactive evolutionary computation method. The KeyGraphs presented in those figure present clusters easy to find and understand, as well as the relations among them provide clear insides about their relations—favoring the chance discovery process.

5 Discussion and final remarks

As discussed in section 3, the goodness of KeyGraph depends on the interpretability of the structures where islands (clusters of high frequency terms connected with links) are connected with bridges (key links). More formally, we define the interpretability as following three criteria: (1) **C1** as the ease to find clusters (except for the meanings), (2) **C2** as the ease to understand the meaning of clusters, and finally (3) **C3** as the ease to comprehend the relations among clusters.

Using these criteria we can characterize and classify the KeyGraphs presented in figures 1, 2, 3, and 4—as table 1 summarizes. For instance, the KeyGraph presented in figure 1(a) is too sparse to recognize any cluster or meaning in the obtained relations. On the other hand, one dense cluster is clearly shown in 2(a) as the result of the small size of the document D . Hence, statistic do not allow the KeyGraph algorithm to output relevant links. Figure 3(b) also presents another interesting behavior. It suggests three meaningful clusters and three tiny ones, however it is difficult to understand and use for chance discovery purposes because they are not connected with each other.

Cyrano’s case revealed that μ becomes high even if the interpretability in terms of **C1**, **C2**, and **C3** becomes worse. This may be the result of the small size of the document D . Under such conditions, the statistics used by the KeyGraph algorithm are not reliable leading to the creation

Table 1: Comparison of the different KeyGraphs generated.

Document	Method	C1	C2	C3	μ
Cyrano	KeyGraph	<i>difficult</i>	<i>difficult</i>	<i>difficult</i>	4.07
Cyrano	KeyGraph+IEC	<i>medium</i>	<i>medium</i>	<i>medium</i>	4.19
Cyrano	KeyGraph+SW+GA	<i>easy</i>	<i>difficult</i>	<i>difficult</i>	10.7
Degas	KeyGraph	<i>medium</i>	<i>medium</i>	<i>medium</i>	1.07
Degas	KeyGraph+IEC	<i>medium</i>	<i>medium</i>	<i>medium</i>	6.18
Degas	KeyGraph+SW+GA	<i>easy</i>	<i>easy</i>	<i>easy</i>	6.44
Marketing	KeyGraph	<i>difficult</i>	<i>difficult</i>	<i>difficult</i>	2.07
Marketing	KeyGraph+IEC	<i>easy</i>	<i>easy</i>	<i>difficult</i>	1.83
Marketing	KeyGraph+SW+GA	<i>easy</i>	<i>easy</i>	<i>easy</i>	6.19

of single clustered KeyGraphs. However, in the Degas and Marketing cases—when enough data is provided to the KeyGraph algorithm— μ increases as the interpretability improves. Another interesting behavior of using μ as the fitness function of a genetic algorithm is the lack of control on the number of clusters to evolve. To tackle this problem, we plan to consider μ and the number of clusters to be evolved.

We also plan to pay attention to the selection mechanism used during the evolution of the KeyGraph’s parameters. The fitness landscape produced by the μ ratio group different KeyGraph topologies under similar values fitness values. This is the result of the normalization terms introduced by the μ ratio. For these reasons, our future work will explore the usage of *nicheing* techniques (Goldberg, 1989) to maintain different KeyGraph parameter settings equally rated under the μ ratio.

In summary, this paper has presented a first attempt to systematize the variation of KeyGraph’s parameters during the creation of KeyGraph for chance discovery. The goal was to create KeyGraphs where (1) the clusters are easy to find, (2) the clusters are easy to understand, and (3) the relations among clusters are easily to comprehend and help in the process of chance identification. The exploration techniques presented in this paper were based on the usage of interactive evolutionary computation and genetic algorithms. Both techniques aimed to create useful KeyGraphs trying to optimize their small world topology. From the experiments reported in this paper we may conclude that the notion of small-world topology contribute to the improvement of the goodness of KeyGraph for chance discovery. Results also showed that enough information needs to be available to create useful KeyGraphs.

Acknowledgments

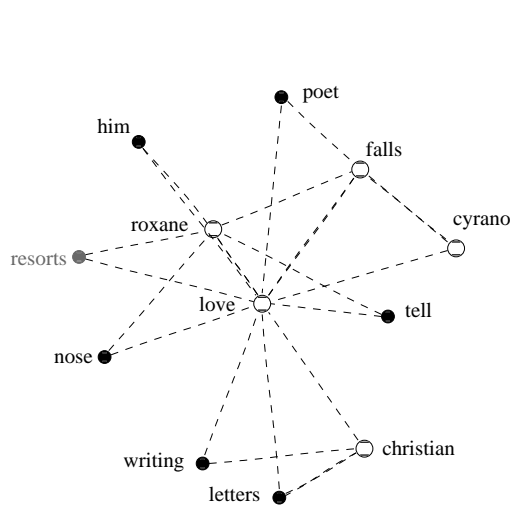
This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-03-1-0129, and by the Technology Research, Education, and Commercialization Center (TRECC), at University of Illinois at Urbana-Champaign, administered by the National Center for Supercomputing Applications (NCSA) and funded by the Office of Naval Research under grant N00014-01-1-0175. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied,

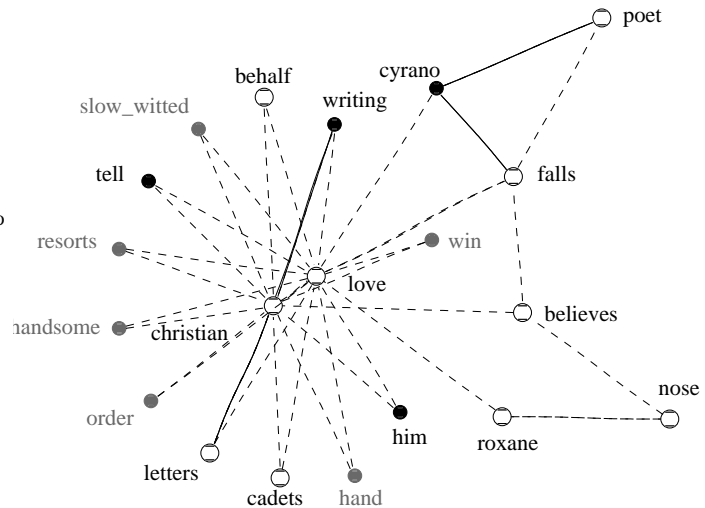
of the Air Force Office of Scientific Research, the Technology Research, Education, and Commercialization Center, the Office of Naval Research, or the U.S. Government.

References

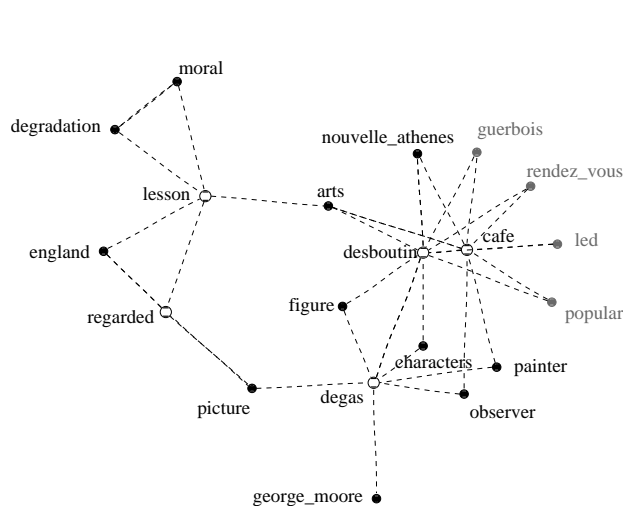
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- Goldberg, D. E. (2002). *The design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publisher.
- Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5), 493–530.
- Goldberg, D. E., Welge, M., & Llorà, X. (2003). *DISCUS: Distributed Innovation and Scalable Collaboration In Uncertain Settings* (IlliGAL Report No. 2003017). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.
- Llorà, X., Goldberg, D. E., Ohsawa, Y., Ohnishi, K., Washida, Y., Tamura, H., & Yoshikawa, M. (2004). *Chances and Marketing: On-line Conversation Analysis for Creative Scenario Discussion* (IlliGAL Report in preparation). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.
- Llorà, X., Ohnishi, K., Chen, Y.-P., Goldberg, D. E., & Welge, M. (2004). *Enhanced Innovation: A Fusion of Chance Discovery and Evolutionary Computation to Foster Creative Processes and Decision Making* (IlliGAL Report No. 2004012). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.
- Ohsawa, Y., Benson, N. E., & Yachida, M. (1998). KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proceedings of Advances in Digital Libraries* pp. 12–18.
- Ohsawa, Y., & McBurney, P. (2003). *Chance discovery*. Springer.
- Porter, M. (1980). An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3), 130–137.
- Rechenberg, I. (1973). *Evolutionstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog.
- Schwefel, H.-P. (1977). *Numerische optimierung von computer modellen mittels der evolutionstrategie*. Volume 26 of Interdisciplinary systems research. Birkhauser Verlag.
- Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9), 1275–1296.
- Walsh, T. (1999). Search in a small world. In *International Joint Conference on Artificial Intelligence (IJCAI-99)* pp. 1172–1177.
- Watts, D., & Strogatz, S. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393, 440–442.



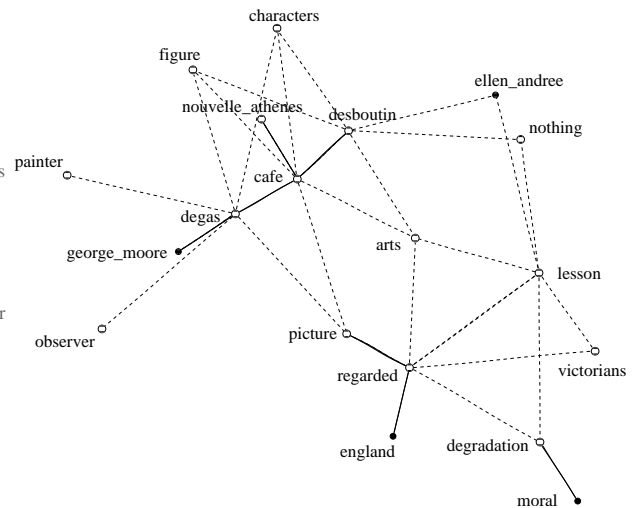
(a) Reviews of Cyrano (default, $\mu = 4.07$)



(b) Reviews of Cyrano (iEC, $\mu = 4.19$)

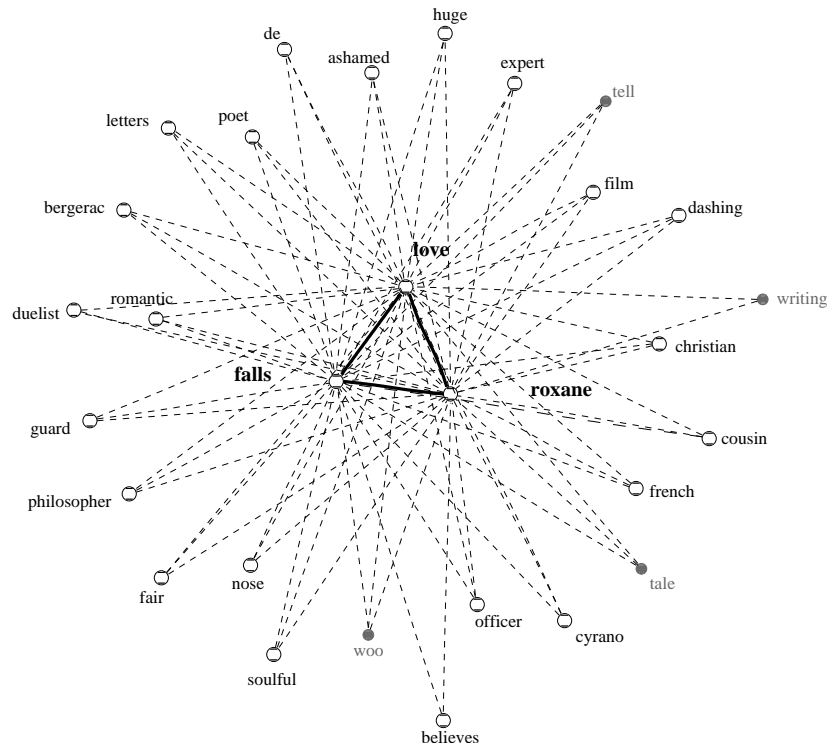


(c) History of Degas' Absinthe (default, $\mu = 1.07$)

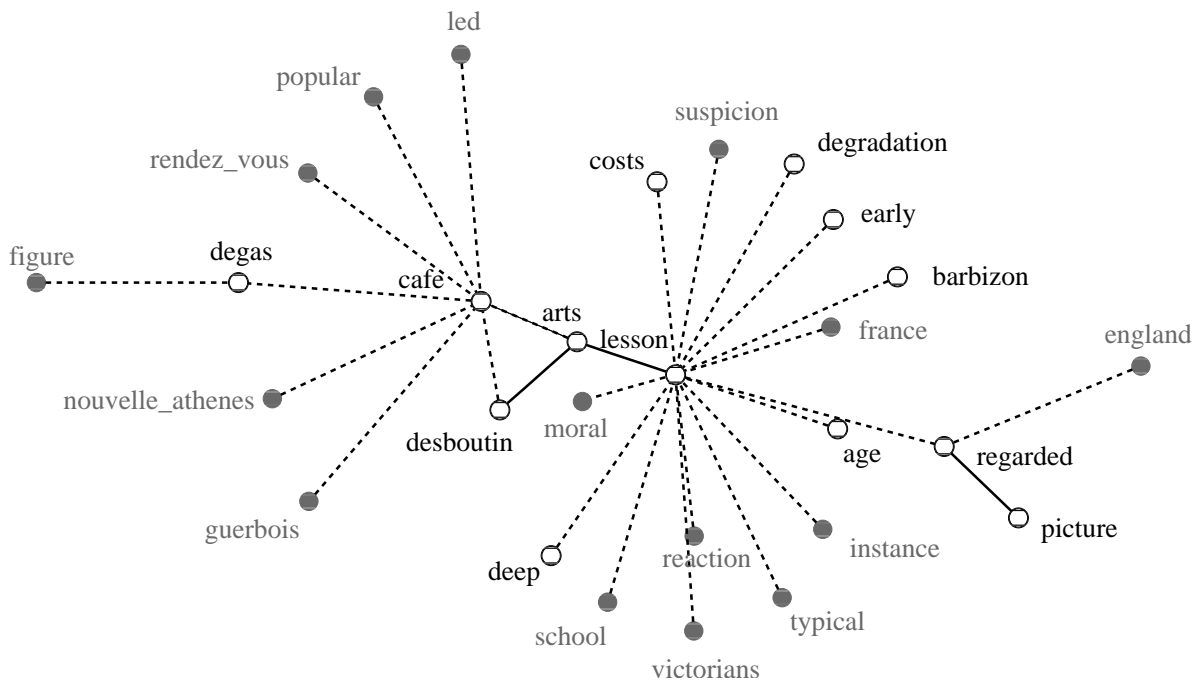


(d) History of Degas' Absinthe (iEC, $\mu = 6.18$)

Figure 1: KeyGraph for the reviews of Cyrano de Bergerac movie and the history of Degas' Absinthe. High frequency terms are displayed as black nodes. Links are represented as solid black edges. Key terms are depicted as gray nodes. Key links are painted as dashed edges. Finally, white nodes represent keywords. Figure compares the KeyGraphs obtained using the default setting of parameters with the ones obtained using interactive evolutionary computation. The evolved KeyGraphs present a better cluster identification, understandability, and relevant chances were easier to find. Moreover, the evolved KeyGraph also presents better small-world measure μ than the ones provided by the default parameter configuration.

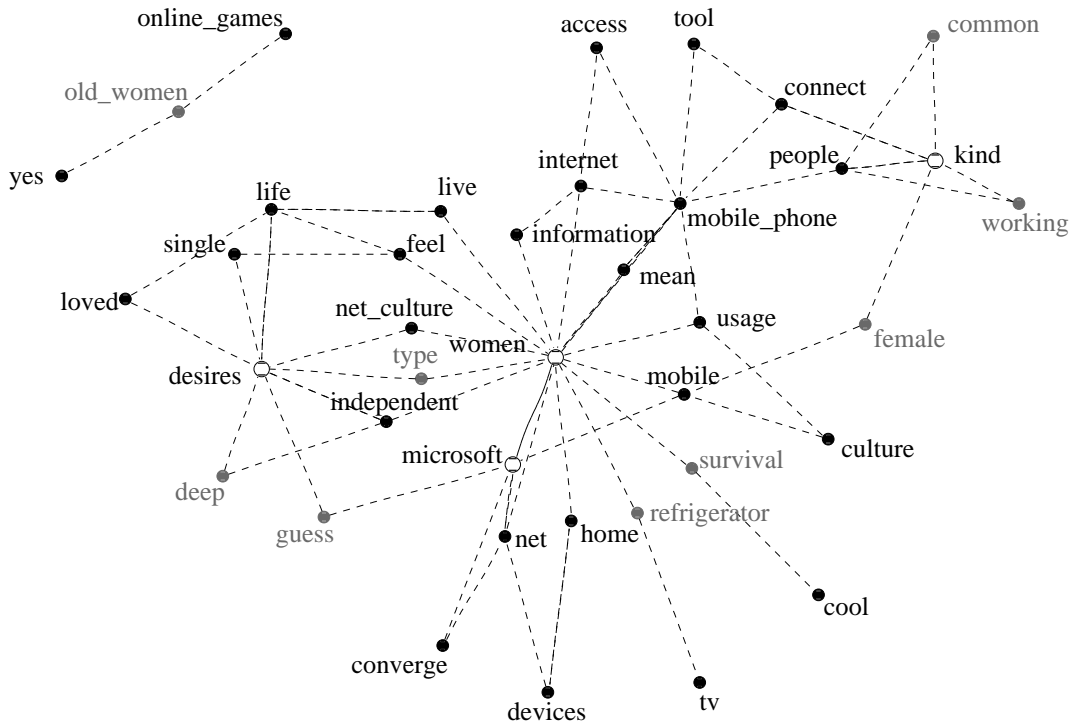


(a) Reviews of Cyrano de Bergerac ($\mu = 10.7$)

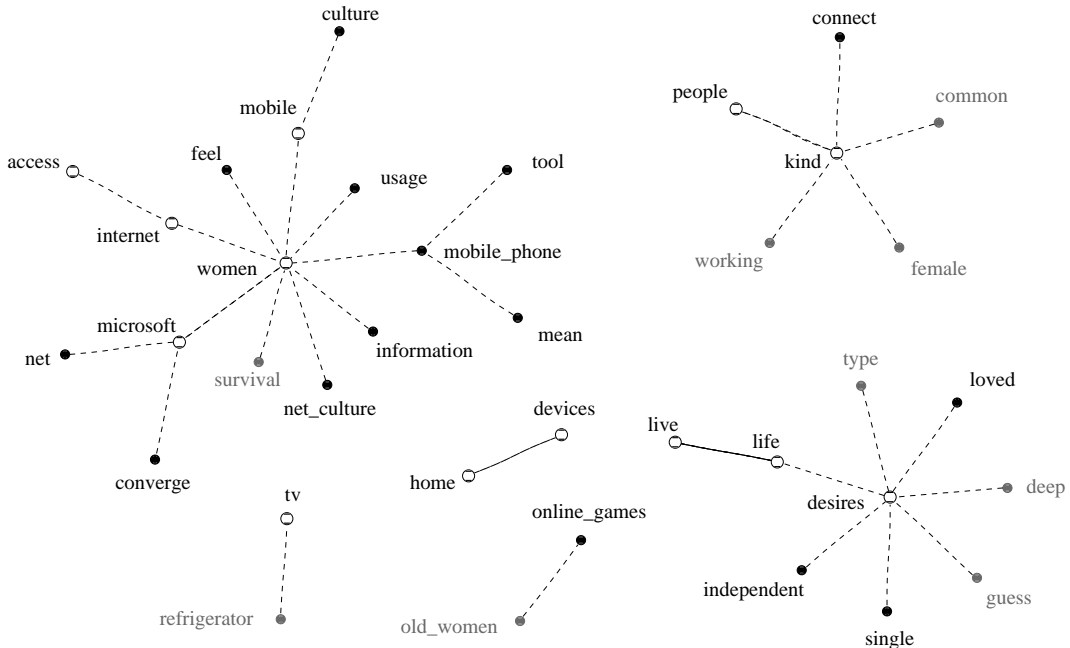


(b) History of Degas' Absinthe ($\mu = 6.44$)

Figure 2: KeyGraph tuned using a genetic algorithm combined with a small-world-based fitness function.



(a) KeyGraph obtained using the default parameters ($\mu = 2.10$)



(b) KeyGraph tuned by means of an interactive evolutionary computation ($\mu = 1.83$)

Figure 3: KeyGraphs of the marketing discussion conducted using DISCUS platform evolved using genetic algorithm combined with a small-world-based fitness function

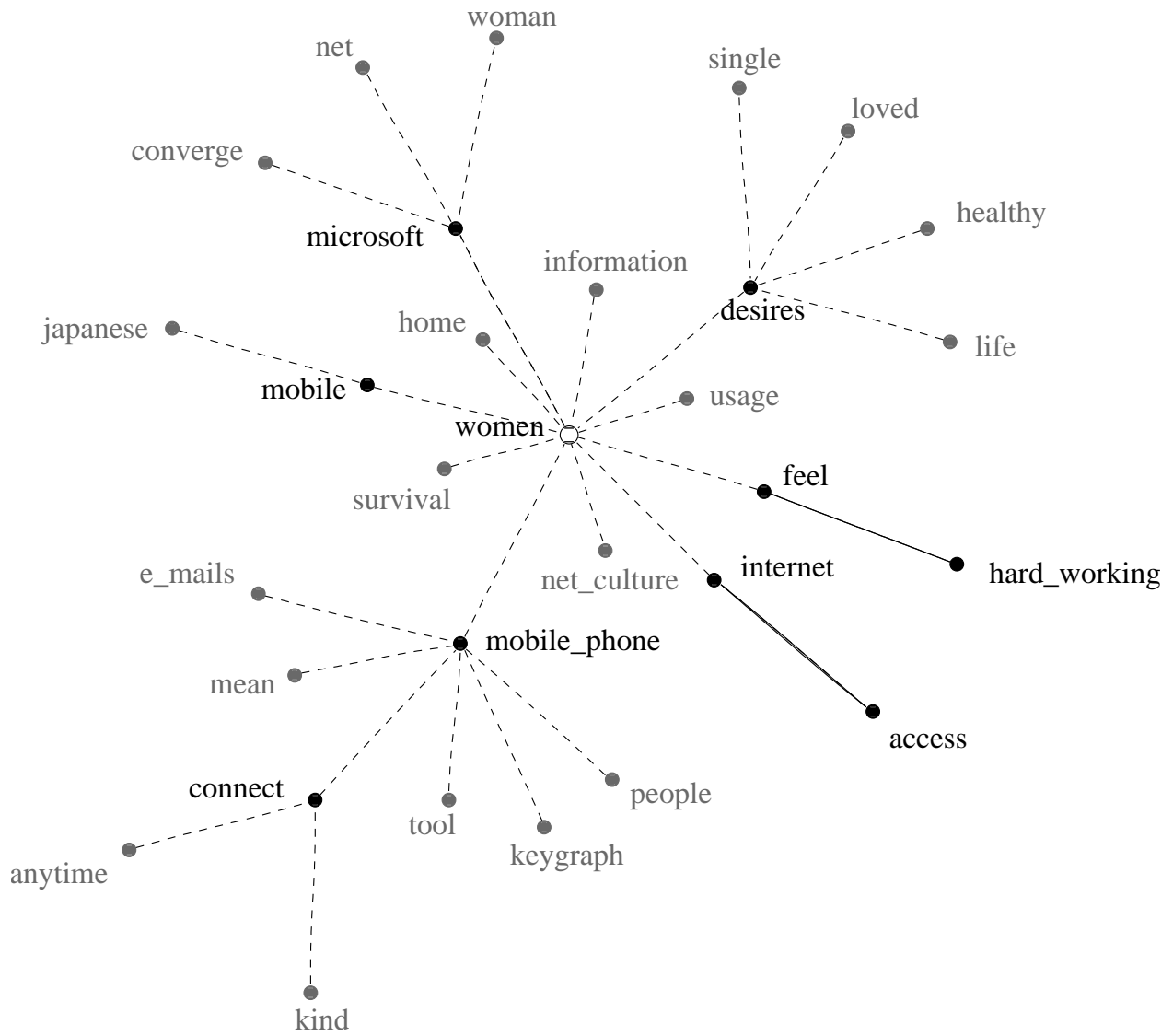


Figure 4: KeyGraph tuned using a genetic algorithm combined with a small world based fitness function ($\mu = 6.19$)