

# Optimizing Preferences within Groups: a Case Study on Travel Recommendation

Fabiana Lorenzi<sup>1,2</sup>, Fernando dos Santos<sup>1</sup>, Paulo R. Ferreira Jr<sup>1,3</sup>, and Ana L. C. Bazzan<sup>1</sup>

<sup>1</sup> Instituto de Informática, UFRGS  
Caixa Postal 15064, 91.501-970 Porto Alegre, RS, Brazil  
{lorenzi,fsantos,prferreira jr,bazzan}@inf.ufrgs.br

<sup>2</sup> Universidade Luterana do Brasil  
Av. Farroupilha, 8001 Canoas, RS, Brazil

<sup>3</sup> Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale  
RS239, 2755, CEP 93352-000, Novo Hamburgo, RS, Brasil

**Abstract.** This work describes a multiagent recommender system where agents work on behalf of members of a group of customers, trying to reach the best recommendation for the whole group. The goal is to model the group recommendation as a distributed constraint optimization problem, taking customer preferences into account and searching for the best solution. Experimental results show that this approach can be successfully applied to propose recommendations to a group of users.

## 1 Introduction

The internet is a rich source of information where users search information about products and services related to their interests and preferences. However, locating the necessary information has become a hard task for the user [7]. Moreover, the information is usually distributed through several locations.

In order to deal with some of these issues, recommender systems have been developed [12]. These systems learn about user preferences over time and automatically suggest products that fit the user needs. Recommender systems are being applied in several domains in e-commerce to suggest products to their customers [2] such as book recommendation (amazon.com) or movie recommendation (netflix.com). The main advantage of recommender systems is the ability to aggregate information and to match the recommendations with the information people are looking for.

Group recommendation is a new challenge to the recommendation area because it is necessary to take into account all members preferences. Each group member elicits his preferences, which means that preferences within the group are not homogeneous. The recommender system needs to aggregate all preferences to formulate a recommendation that suits the whole group.

Several group recommender systems have been developed in the past years, in different domains. *Let's Browse* [3], for instance, is a group recommender system which recommends web pages to a group of users who are browsing the web.

Another example is *MusicFX* [9], a system used in a fitness center to adjust the selection of background music to best suit the preferences of people working out at any given time. A special feature found in this system is that a group is composed by people who happen to be in the place at the same time. *MusicFX* uses explicit preferences of all participants to make a music selection that will be listen by everyone who is present. In this case, the group is composed by strangers rather than family members or friends.

*Intrigue* [1] is another example of group recommender system. The system recommends attractions and itineraries by taking into account preferences of heterogeneous groups of tourists (such as families with children) and explains the recommendations by addressing the requirements of the group members. Attractions are separately ranked by first partitioning a user group into a number of homogeneous subgroups with the same characteristics. Then each subgroup may fit one or more stereotypes and the subgroups are combined to obtain the overall preference, in terms of which attractions to visit for the whole group.

The group recommendation task may become more difficult according to the complexity of the domain. Recommendation of travel packages, for instance, is composed by several information components such as flights, hotels, and attractions [13] [5]. Besides the specific knowledge necessary to assemble all the components, each user has different preferences that need to be considered during the recommendation process. A group member may prefer flying during the day and staying in a four-star hotel; while another member prefers flying at night and staying in a hostel to save money. Besides the fact that group members have different and hard criteria to make decisions, sometimes group members do not want to let other members know their preferences. In these cases, the privacy is very important and the system cannot allow users to see each other preferences. Thus, finding the best option for the entire group according to the preferences of each individual and keeping the privacy of each member is a tricky task. The system needs to find the global optimal recommendation in a distributed fashion.

The formalism of distributed constraint optimization problem (DCOP) has been proposed to deal with the problem of coordinating and optimizing agents' interactions. It has been used in multiagent systems (MAS) in several domains [8] [10] [11]. DCOP is related to the constraint satisfaction problems (CSP), a well known technique in AI, in the sense that it deals with assignment of values to variables under certain constraints. However, DCOP is more difficult. First, it deals with optimization (not only with satisfaction) meaning that the best solution must be found (not any one). Second, the assignment is computed in a distributed way.

CSP was also used in [6], where authors proposed a multiagent recommender system to arrange meetings for several participants taking into account constraints for personal agendas. Three different agents were proposed: the *personal assistant agent* is the interface between the user and the MAS; the *flight travel agent* is connected to a database of flights; and the *accommodation hotel agent* is responsible for finding an accommodation on the cities involved in the meeting. However, the system does not provide the best recommendation to the group,

due to the limitations of the CSP formalism (any recommendation is acceptable instead of the best).

We propose a DCOP-based multiagent recommender system to perform the travel group recommendation task. Aspects of travel group recommendation task such as customers, their preferences, and the need of coming out with a recommendation for the whole group can be viewed as components of a DCOP instance. Each user is represented by an agent that is responsible for negotiating with a recommender agent who holds the information about the travel services, trying to get the best recommendation according to customers.

Experimental evaluations were performed to verify the feasibility of the DCOP-based multiagent recommender system. Test cases were generated and solved with a DCOP algorithm, namely the *distributed pseudotree optimization* (DPOP) [11]. The results obtained have shown that using DCOP to optimize the problem of finding a recommendation to a group is feasible both in terms of running time and communication load.

The rest of the paper is organized as follows: section 2 describes the DCOP framework. Section 3 shows the proposed multiagent recommender system, providing details about the use of DCOP to specify the travel groups recommendation process. Section 4 discusses the feasibility of the proposed multiagent recommender system using the DPOP algorithm. Section 5 summarizes the contributions and shows future direction for this work.

## 2 Distributed Constraint Optimization Problem

DCOP represents a generic framework for the resolution of distributed problems with a significant application in MAS. The challenge is to find the best distribution and value attribution of a set of variables to a set of agents with interdependencies. In a DCOP, differently from a distributed constraint satisfaction problem (DisCSP) [14], the interest is to optimize the restrictions and not only to satisfy them. DCOP is associated with a global function and the objective is to maximize or minimize it. This function depends on a cost value associated to each restriction.

The approaches for dealing with DCOP in real life problems should consider that the agents must be able to optimize the global function in a distributed way, using only local communication. It is not acceptable to use a central agent responsible for all the processing. Also a DCOP algorithm should be capable of finding the solution with the agents working in an asynchronous way. Finally, the approach should provide quality guarantees.

DCOP is a formalism to model a range of agents coordination issues. A DCOP consists of  $n$  variables  $V = \{x_1, x_2, \dots, x_n\}$ , where each can assume values in a finite, discrete domain  $D_1, D_2, \dots, D_n$ . Each variable is assigned to one agent that has the control over the values of the variable. The goal of the agents is to choose values for the variables to optimize a global objective function. This function is described as an aggregation over a set of cost functions related to pairs of variables (in the case of binary constraints). Thus, for each pair of variables

$x_i, x_j$ , there is a cost function defined as  $f_{ij} : D_i \times D_j \rightarrow \mathbb{N}$  [10]. A DCOP can be represented by a constraint graph, where vertices are variables and edges are cost functions between variables.

Although some algorithms have been proposed to deal with DCOP, here we use the DPOP [11] algorithm. It must be said that we have also used other algorithms such as Adopt [10]. Due to lack of space we do not show the results here but notice that DPOP has performed better than the others in this setting. DPOP uses a message-passing scheme to allow the communication among the agents and to achieve the solution in a distributed fashion. DPOP is based on the dynamic programming technique. It provides solutions quickly with a low number of messages since it is not fully asynchronous. On the other hand, it requires an exponential memory space. It is a complete algorithm, i.e., it always finds the optimal solution.

### 3 The Multiagent Recommender System

In our DCOP-based multiagent recommender system, a community of agents share a common goal (the travel package recommendation) as well as individual goals (the individual preferences). The proposed multiagent recommender system is composed by two different kind of agents: the user agent (UA) and the recommender agent (RA). The UAs work on behalf of each user and represent their travel preferences. Each UA knows the user individual preferences for each travel service (hotel, flight companies, tour operators, etc). In the recommendation process, these agents interact with the recommender agent to reach the global optimal recommendation. The RAs work on behalf of suppliers of travel services and each one has information about a different kind of service. Each RA is responsible for collecting information from a supplier. This step is reported elsewhere [4].

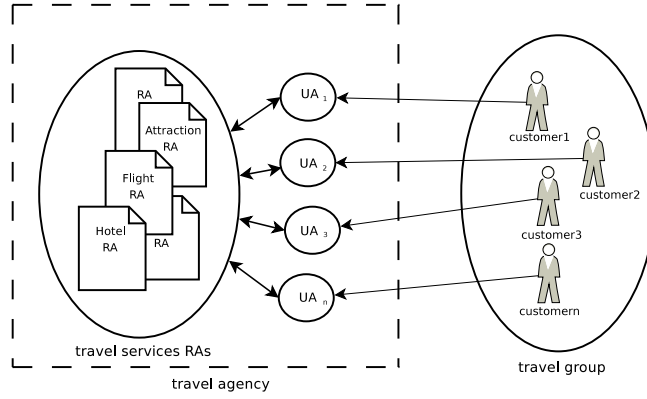
Figure 1 illustrates the scheme of the proposed multiagent recommender system. First, it is required that each user set up his preferences and inform his UA. After, the UAs start an interaction with the RA to get a list of possibilities for the particular service. This list is ranked according to the user preferences for this service. It is important to mention that a travel package recommendation is composed by several components (flight, hotel, attractions and so on). However, we are considering in our example (shown in Figure 2) just one service (e.g. hotel) to recommend.

The major challenge in this approach is to guarantee the best recommendation for the whole group of users. The recommendation process is easy when it deals with an individual user. However, when we have a group of users, the system has to find the best recommendation for the whole group, i.e., the system has to take into account all preferences of the users .

Group recommender systems reported in the literature are based on different approaches in what regards how to reach to the final decision. These approaches were already discussed in section 1. Here we revisit them to draw comparisons with our approach. In *MusicFX*, for example, the system selects and plays music

autonomously based on the preferences of the group members who are present in the gym. In *Let's Browse*, there is a group member responsible for making the final decision. This member decides the best option for the group. *Intrigue* uses a similar approach, but it is the tour guide who decides which tour should be taken by the group. Another existing approach is to assume that group members will arrive at the best decision through conventional discussion. In this case, group members could chat or exchange e-mails, trying to reach a consensus.

As we propose a multiagent recommender system to recommend travel packages to groups, in our point of view, none of the approaches mentioned is adequate to our scenario. First, a travel package has different components and a group has several members. The group would take too long to make a final decision. Second, the approach based on a member who is responsible for the final decision is a centralized one.



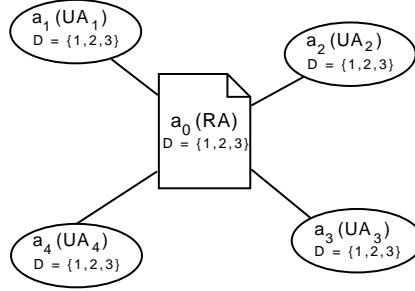
**Fig. 1.** DCOP-based multiagent recommender system

As mentioned, each UA represents one user that ranks all information obtained from the RA.  $\mathcal{R}$  is the set of possible recommendations defined as  $\{r_1, r_2, \dots, r_{|\mathcal{R}|}\}$ . The individual ranking  $S_a(r)$  for the set  $\mathcal{R}$  is defined as follows: we compute the preference  $P_a(r)$  of UA  $a$  regarding a recommendation  $r$ . Assuming that there is a permutation of  $\mathcal{R}$  such that  $P_a(r'_1) > P_a(r'_2) > \dots > P_a(r'_{|\mathcal{R}|})$ , we defined an individual ranking  $S_a(r)$  for the set  $\mathcal{R}$  as  $S_a(r'_1) = 0, S_a(r'_2) = 1, \dots, S_a(r'_{|\mathcal{R}|}) = |\mathcal{R}| - 1$ . The preference values  $\{0, 1, \dots, |\mathcal{R}| - 1\}$  are used to simplify the notation and save space in the description. However, any other form of preferences representation (e.g. a table) could be used. These values can be any positive integers and the most preferable recommendation must have the lowest value.

We map the UA rankings into cost functions and these cost functions are aggregated to describe the global objective function. The optimum of this ob-

jective function provides the best recommendation of a travel package to the group.

For each travel group, the DCOP is defined as follow: a set of variables  $\{a_0, a_1, \dots, a_n\}$ , where the variable  $a_0$  is assigned to the RA and  $a_i$  is assigned to the  $UA_i$  (where  $0 < i \leq n$ ). These variables can take values from the domain  $D = \mathcal{R}$ . For sake of illustration, Figure 2 shows a simple example of a DCOP of a travel group recommendation, with four UAs  $\{a_1, a_2, a_3, a_4\}$  and one RA  $\{a_0\}$  that is representing hotels information. This RA has three option of hotels that can be recommended to the group, thus the domain  $D$  of each variable is  $\{1, 2, 3\}$ . Each UA has a ranking of these hotels according to its user preferences.



**Fig. 2.** The DCOP for one group recommendation case.

The cost function for each relation between one UA and the RA is computed based on the ranking  $S_a(r)$  for the set  $\mathcal{R}$  of recommendations. Thus, there is one cost function  $f(a_0, a_i)$  to represent the individual ranking between each pair  $(a_0, a_i)$  for all  $0 < i \leq n$ . Each of these cost functions has the following form

$$f(a_0, a_i) = \begin{cases} S_{a_i}(r) & \text{when } a_i = r \text{ and } a_0 = r \\ \infty & \text{otherwise} \end{cases} . \quad (1)$$

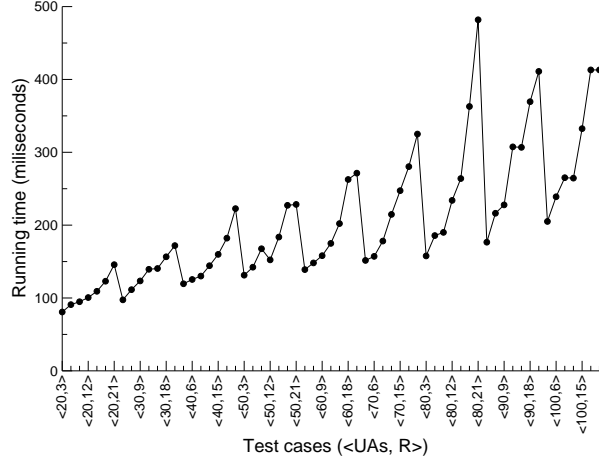
The first case of Equation 1 captures the situation where the agents  $a_0$  and  $a_i$  agree on the recommendation. The second case captures the situation where some agent disagrees with the RA. Equation 2 shows the global objective function, which is the aggregation over all cost functions.

$$\mathcal{F} = \sum_{i=1}^n f(a_0, a_i) . \quad (2)$$

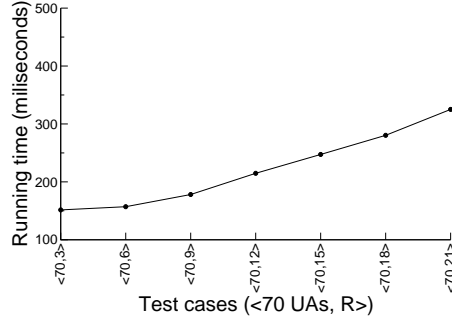
Since the individual ranking is defined in an ascendant way, the solution to the DCOP (and thus the best recommendation to the travel group) is the one that minimizes this global objective function. It is important to note that our DCOP-based multiagent recommender system can be extended to have more than one RA.

## 4 Experimental Evaluation

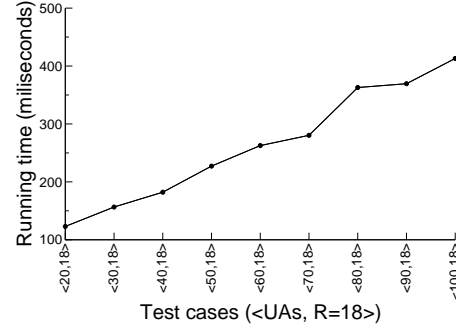
We have conducted a set of experiments to empirically evaluate the feasibility of the proposed multiagent recommender system. We use two measures to evaluate the performance:



(a)



(b)



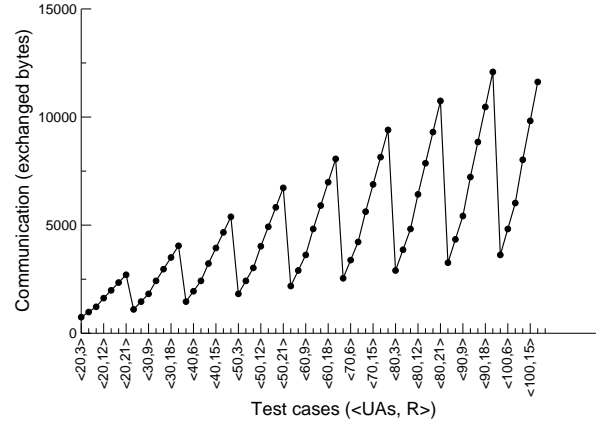
(c)

**Fig. 3.** (a) Running time, in milliseconds, for each test case. (b) Details of the nearly linear running time in the domains size for a fixed number of 70 UAs. (c) Details of the linear running time in the number of agents for a fixed  $\mathcal{R}$  with 18 recommendations.

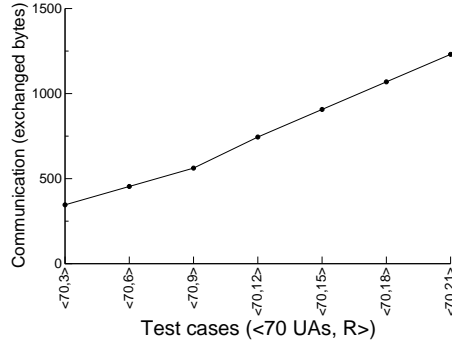
- *Running time*: time required by the algorithm to solve the recommendation for the group taking into account user preferences.

- *Communication (network usage)*: number of bytes transferred through message exchanges among the agents during the algorithm execution.

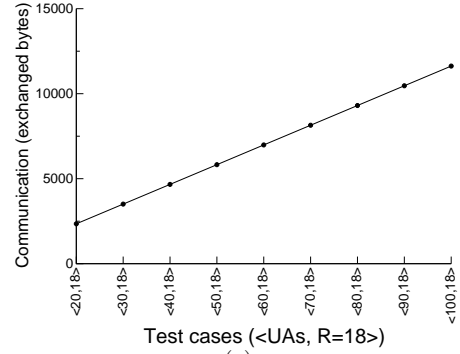
Test cases have 20, 30, 40, 50, 60, 70, 80, 90, and 100 agents, each representing one individual in the travel group (one UA). In addition, we have one RA. For each of these cases,  $\mathcal{R}$  takes the following values: 3, 6, 9, 12, 15, 18, and 21 (representing possible recommendations).



(a)



(b)



(c)

**Fig. 4.** (a) Communication (sum of exchanged bytes) for each test case. (b) Details of the linear communication in the domains size for a fixed number of 70 UAs. (c) Details of the linear communication in the number of agents for a fixed  $\mathcal{R}$  with 18 recommendations.



In each experiment the data was averaged over 20 runs. We run the original implementations of the DPOP algorithm, which are available on the internet<sup>1</sup>.

Figure 3(a) shows the running time in milliseconds for each test case. We can see that the running time is nearly linear in the number of UAs and in the domain ( $\mathcal{R}$ ) sizes. To show with more details these increasing rates, Figure 3(b) shows how the running time is nearly linear in the domains size for a fixed number of 70 UAs and Figure 3(c) shows the running time for an increasing number of UAs, for a fixed  $\mathcal{R}$  with 18 recommendations. The algorithm provides very satisfactory results regarding these metrics and proves the feasibility of using DCOP in the group recommendation task. In the hardest test case, with 100 UAs and 21 travel recommendation options the algorithm takes only about 413 milliseconds to solve the DCOP and provides the optimal recommendation to the group.

Figure 4(a) shows the total number of bytes transferred. The results are both nearly linear in the number of UAs and in the domain ( $\mathcal{R}$ ) sizes. Figure 4(b) shows details of the increasing in the domains size for a fixed number of 70 UAs while figure 4(c) depicts the linear growing rate in the number of UAs for a fixed  $\mathcal{R}$  with 18 recommendations. For example, in the case of 100 UAs and 18 domain values, the total number of bytes transferred is about just 11625 bytes.

## 5 Conclusion and Furure Work

Recommender systems are been used in e-commerce to help users to find out better products according to their needs and preferences. Multiagent recommender systems use agents to help in the solution process, trying to improve the recommendation quality. Agents cooperate in order to reach the best recommendation for the whole group. The recommendation for group of users is considered a challenge to recommender systems due the fact that is not an easy task to find out the best solution that satisfies all the users in the group.

In this paper we have shown that the DCOP formalism can be successfully used to help the group recommendation process. Agents work on behalf of the users and try to get the best recommendation according to the user preferences. Experimental evaluations have shown the feasibility of using the DPOP algorithm to solve the DCOPs generated for each recommendation case.

As a future work we intend to adapt and test our system to recommendations where the preferences are interrelated. These interrelation appears for instance when the users define the preferences for a travel option A as a result of the available travel options B.

## References

1. L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices.

---

<sup>1</sup> DPOP is available at <http://liawww.epfl.ch/frodo/>

- Applied Artificial Intelligence: Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, 17(8-9):687–714, 2003.
2. A. Goy, L. Ardissono, and G. Petrone. Personalization in e-commerce applications. In *The Adaptive Web*, pages 485–520. Springer Berlin / Heidelberg, 2007.
  3. H. Lieberman, N. W. V. Dyke, and A. S. Vivacqua. Let’s browse: A collaborative web browsing agent. In M. T. Maybury, editor, *International Conference on Intelligent User Interfaces - IUI99*, pages 65–68, New York, 1999.
  4. F. Lorenzi, A. L. C. Bazzan, and M. Abel. Truth maintenance task negotiation in multiagent recommender system for tourism. In *Proceedings of the AAAI Workshop on Intelligent Techniques for Web Personalization and Recommender Systems in E-Commerce (AAAI 2007)*, pages 122–125, July 2007.
  5. F. Lorenzi, D. S. Santos, D. de Oliveira, and A. L. C. Bazzan. *Task allocation in case-based recommender systems: a swarm intelligence approach*, pages 268–279. Information Science Reference, 2007.
  6. S. Macho, M. Torrens, and B. Faltings. A multi-agent recommender system for planning meetings. In *Workshop on Agent-based recommender systems (WARS’2000)*, 2000.
  7. P. Maes. Agents that reduce work and information overload. *Commun. ACM*, 37(7):30–40, 1994.
  8. R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 310–317, Washington, DC, USA, July 2004. IEEE Computer Society.
  9. J. McCarthy and T. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In M. T. Maybury, editor, *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW’98*, pages 363–372, Seattle, 1998.
  10. P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 161–168, New York, USA, 2003. ACM Press.
  11. A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In L. P. Kaelbling and A. Saffioti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 266–271, Edinburgh, Scotland, August 2005. Professional Book Center.
  12. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings ACM Conference on Computer-Supported Cooperative Work*, pages 175–186, 1994.
  13. F. Ricci, D. Cavada, N. Mirzadeh, and A. Venturini. Case-based travel recommendations. In D. R. Fesenmaier, K. Woeber, and H. Werthner, editors, *Destination Recommendation Systems: Behavioural Foundations and Applications*, pages 67–93. CABI, 2006.
  14. M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems*. Springer, Berlin, 2001.