AI Communications 0 (0) 1
IOS Press

A Genetic Algorithm for Group Formation in Elderly Communities

Juan M. Alberola [a,*] , Elena del Val [a] , Angelo Costa [b] , Paulo Novais [b] and Vicente Julian [a]
[a] *D. Sistemas Informáticos y Computación, Universitat Politècnica de València, Valencia, Spain*
E-mails: jalberola@dsic.upv.es, edelval@dsic.upv.es, vinglada@dsic.upv.es
[b] *Centro ALGORITMI, Escola de Engenharia, Universidade do Minho, Guimarães, Portugal*
E-mails: acosta@di.uminho.pt, pjon@di.uminho.pt

Abstract. Over the last few years, agent technology has proposed different approaches for the development of systems that support the formation and dynamic management of teams or groups. Following this idea, this paper proposes a model for group formation in elderly communities defining it as a Coalition Structure Generation Problem. Specifically, the proposal has been used to analyse which is the best way to organize older people into activity groups in elderly communities. The model parameters are physical requirements, preferences and social relationships, being the model able to learn from each execution and improve the future configurations. The results show near-optimal solutions to all proposed scenarios, beating greatly the computational time of CPLEX, and flexibility to deal with different scenarios and configurations.

Keywords: Genetic algorithm, group formation, elderly activities

1. Introduction

There is an inherent value in group formation. Society has built itself to benefit from grouping of people (socially and otherwise), and often some of the greatest achievements are done by groups of people. Thus, collaboration is essential for successfully achieving any type of goal.

We can observe the growing importance of the concept of business teams in the literature. But this is not the only field where teams, groups, coalitions or partnerships are being used. For instance in education, educational organizations have shown a growing interest in shifting towards teaching paradigms that promote teamwork [1–3]; in sports competitions, it is well know the importance of building and managing a team to achieve success; and in other fields, such as science, many of the most important results arise from the formation of groups and their collaboration.

One issue with grouping people is the human character. The likeness of each person towards others varies greatly producing matches and mismatches. This are the reasons why some groups do not work and tend to disband. This is true to every type of groups, and to any person's age, gender and race. Group formation is complex and socially demanding, thus it is a relevant domain to today's society.

In general, any task with hints of complexity require the collaboration of more than one individual. It is essential that the current technology has the ability of giving support to the needed processes of formation and management of groups or coalitions with the aim of maximizing the utility or expected benefit. Furthermore, given the nature of the computer systems, calculating time-expensive and complex algorithms is easy to them.

In this sense, agent technology, although still immature in some ways, allows the development of systems that support the formation and dynamic management of these teams. Many tasks cannot be completed by a single agent because of limited resources or capabilities, and even if the task can be done by a single agent, the performance may be too low to be acceptable. In these situations, agents may form groups to solve the problem or accomplish the task by cooperation. This work is focused on the formation of teams in order to do some specific task.

One social domain that uses group formation daily and requires a large social effort is elderly care. Elderly people that are residing at a caregiving institution or using caregiving services often interact with other people through group activities. Caregivers have

---

[*] Corresponding author. E-mail: jalberola@dsic.upv.es.

reported that managing daily activities is one of the hardest tasks they have due to the health and personal requirements each elderly person has [4, 5]. This is a common issue to most caregivers, and while there are solutions for this issue (like manual planning) they are often time-consuming and non-optimal.

Our aim is to use this a novel approach of group formation using genetic algorithms and use it in the domain of geriatric assistance.

Different studies [6–13] have shown the benefits of a constructive group activities programme for the elderly, increasing their happiness levels and wellbeing. In most elderly communities exists the figure of the activities manager, typically assumed by a psychologist or a registered nurse, whose task is to create a list of activities that please the elderly communities' users (care-receivers). An usual issue is the lack of novelty and low significance of the events, leading to bored and unengaged care-receivers. Caregivers should be able to arrange activities that relate to the care-receivers, e.g., likes, health condition, background, social attachment. Exploiting the social interaction is key to keep a harmonious environment, thus it is essential to please the largest number of care-receivers [14]. The issue is that finding activities that please everyone is rather difficult and most of the activities that do that are ones that require little effort by the care-receiver (like watching television) defeating the goal of promoting active aging through playful psychological and physical activities.

One possible solution is to divide the community into groups, taking into account the relationships between the participants (tastes, health status, friendship). In addition, appropriate activities can be proposed to the groups in order to optimize the overall satisfaction of the community. The issue with this solution is that it is a hard problem to find suitable associations between the users. For instance, from the three areas (likes, health condition, friendship) the values can range from love to hate, thus one care-receiver may love another but hate every activity that the other likes, which may eventually lead to unhappiness of the care-receiver in the long term. By using technological helpers the task can be eased by performing the grouping task.

This idea emerged from an issue encountered in the iGenda platform [15, 16]. When in a specific environment where the care-receivers were forced to interact with other care-receivers the iGenda was not able to provide events suggestions in a fashionable time. The complexity of the task and specificity was not initially foreseen thus the iGenda was not designed for this task.

Traditionally, allocating agents into optimal groups has been a field of study for coalition formation [17–19]. Many coalition formation algorithms focus on optimally dividing coalitional payoffs [20–22], which are the resulting benefits from carrying on a task as a group.

Genetic Algorithms (GA) have also contributed to the state-of-art in group formation. They consist in general optimization and learning algorithms based on the evolutionary processes found in the nature. Candidate solutions for a problem form the genetic population of the algorithm, which gradually converges towards high quality solutions by applying genetic operators like mutation and crossover. GA can be used as an implicit learning and adaptation mechanism in environments where dynamic and structure is also uncertain. This is perhaps what makes GA an adequate approach to group formation problems, since they can be used to learn and adapt both to the different needs and goals of the group's members. Moreover, their simple approach to the optimization means that the time consumption is low in each execution, thus for the same time that classic coalition systems take to provide one optimal answer, a GA can provide multiple outputs.

In this sense, this work proposes the use of GA in order to solve a specific problem of group formation. Concretely, the proposal has been used to analyse which is the best way to organize older people into activity groups in elderly communities (e.g. nursing homes, day-care centres).

This paper is structured in the following way: section 2 presents the proposed model and the problem definition; section 3 presents the genetic algorithm design, with equations and examples that help to envision the development; section 4 present the experiments and the results of 4 scenarios execution; and finally section 5 present the conclusions.

2. Related work

Different approaches have tried to deal with the problem of team formation as a decision support system in many diverse domains. We have analyzed different tools that try to employ intelligent coalition formation techniques for humans. For a more detailed survey about the development of algorithms and mechanisms for coalition structure generation, please see [23].

One of the most well-known approaches is the work presented in [17] which dealt with the problem of coalition formation and cost sharing in a group-buying electronic marketplace. The proposed tool allowed buyers to form coalitions in order to take advantage of volume based discounts.

More recent works have built solutions in many different domains. In [24], authors describe a computerized coalition formation decision support system that attempts to form a coalition to create a government that is expected to be durable and stable. The approach employs theories from political science and utilizes insights from artificial intelligence to build a tool.

Another example is the work proposed in [25]. This work proposes a solution for the social ride sharing problem, where a set of users, connected through a social network, arrange one-time rides. The tool solves the optimization problem of forming groups of car occupants to minimize the travel cost of the overall system. TeamGen [26] is an interactive team formation system to form project teams interactively by leveraging professional social network information of potential members. The tool has a web-based interface and employs an iterative algorithm to form hierarchically structured project teams in an interactive way.

Recently, [27] tries to propose an approach for the formation of R&D project teams, the proposed algorithm selects the most expert individuals facilitating the knowledge sharing among people. Finally, in the educational context, [28] proposes a tool which aim is to cover the team formation process in class improving different teamwork aspects such as team dynamics and student satisfaction. Another approach is Bang [29] which is a system that selects exercises for students training providing tools to evaluate the student performance and the expected performance of possible teams. This is done in the context of programming competitions.

Nevertheless, there are few academic works, similar to those described, in the area of health. Software and services directed to professional caregivers concentrate on private companies due to the structure of health service operators. Tools specific to decision support system and personal health records are presented next, in an effort to show the available features and how they compare to our developed tool.

2.1. Hometeam

Hometeam (https://www.hometeamcare.com) is a company that produces a product with the same name. Hometeam is divided in to two parts, the software and the service. The idea behind the product is to provide health care and, at the same time, provide that information for the relatives of that people. They aim to provide care like a relative would and having those relatives remotely evaluating the service provided. Furthermore, the software intelligently combines families with seasoned caregivers, connects families with the care-receivers through mobile technology, and helps caregivers to plan activities that improve the wellbeing.

In terms of the service, the company hires caregivers to provide the caring service and to attend the specific health requirements that the users have. They focus on three pillars: quality of life, companionship, personal care and medical assistance.

The software is used as an activity tracker and user manager directed to the caregiver. The caregiver inputs all information related to the care-receiver activities (what has the user done or not, if it was planned) and his/her activities around the home. The information is structured in a blog-like environment for easy consultation, where the relatives and medical staff can comment on each activity, providing useful insight. Although it does not have a automatized decision process, it presents statistics and logs that may be used to support the decision made by a relative, caregiver or doctor. Furthermore, the software component serves as a personal health record system, that tracks the care-receiver health information, making it available to the relevant players. Moreover, it also distinguishes itself from having its caregivers identify activities that each patient prefers through a Likert ranking system (in [30] you can find a brief explanantion about Liker scales).

2.2. Tabula Pro

The Tabula Pro software (https://www.tabulapro.com) was developed and designed to manage information of the care-receivers and have it automatically alerting the caregivers about the care-receivers documentation. It provides a research report that provides aggregated information about the care-receivers. The software is designed to be a complete management system with features such as: user security leveling, visual interfaces, care-receiver management, caregiver management, alerts, tracking and reports.

The software has multiple levels that assure the correct access to the system for each user. Meaning that the quantity of features increases with the clearance level of the accessing user. This enforces the usage of their personal health record to be used by all staff and to be updated often. The visual interfaces provide information about the personal health record as well as the care-receivers calendar and direct contacts. This way the caregivers can easily view and manage the activities that an care-receiver can perform based on their health status.

The resident management is a personal health record system that is updated by the caregivers. The task of updating the records is triggered by the alert feature that monitors each care-receiver record and if exceeds a predetermined date the system will alert the responsible caregivers of this event.

The caregiver management follows the concept of the resident management, where the difference resides in the quantity of information and the type of information. Mainly, it serves as a time manager to keep track of the staff shifts and care-receiver requirements, suggesting work calendars for the employees.

Lastly, the tracking and reports are used to view the care-receivers requirements and medical evolution, providing additional information for the decision of attributing the caregiver that has experience with that medical condition.

2.3. *Resident Management System*

The Resident Management System (https://www.ltcquest.com/v11_rms.asp) is a complete suite for management of a residency. It focuses on the management of electronic health records and information about the caregivers. It also provides relevant information to the caregivers about the medication they have to take including images that clearly identifies them.

Overall it is just an administrative tool that is designed to be used by mostly informal caregivers to keep track of the care-receivers, manage their daily activities, like exercise and medication intake, and maintain their health data so it can be monitored and revised by formal caregivers and doctors.

2.4. *Sagely*

Sagely (http://www.gosagely.com) is directed to informal and formal caregivers that want to monitor the care-receivers though technological means. The most important features are: calendar management, media display and edit, newsletters, and a community and family mobile application.

The calendar management is designed to support caregivers and residencies, to plan a full month of activities. While not very modular, this feature produces printable calendars that may be distributed to the caregivers and care-receivers, so they can see the overall month planning and chose to perform those activities or other ones.

The media display and edit feature is directed to the car-receivers. It is an application that uses Smart TV's to display activity calendars, announcements, photos, birthdays, food recipes, special events. The objective is to use an appliance that the care-receiver has and knows to operate to present the platform information. Furthermore, the visual aspect can be personalized by the care-receivers by creating a slideshow of images that are shown when the user is not interacting with the TV.

The newsletters are distributed by email or in paper form and serve as promotion of events and active aging. The platform provides a tool to build the newsletter with information present on the platform and adding new information, resorting to established templates that format them automatically.

The community and family mobile application are the most innovative feature, being designed to support the caregivers (formal and informal). The community application monitors the activities of the care-receivers and their participation in activities and interaction with the TV. It produces a report that shows the evolution of the care-receiver and their level of interest (low levels may represent a health problem or loss of interest in the current suggestions). Furthermore, the application has a sharing function that is able to share those reports or pictures with the care-receivers' family. The family application shows the care-receiver calendar and notification of administrative processes, like authorizations and changes on medication; it also serves as a social network, where the family members can interact with each other and share information, suggestions and photos.

These platforms show that there is a large investment in terms of solutions to monitor and care for elderly people, whether they are at home or at residencies. Most of them are almost purely administrative, and easy the bureaucratic process that is keeping the personal health record updated as well as keeping track of the activities and general wellbeing of the care-receivers.

From our observations it was clear that these applications require that the caregivers spend a large amount of time interacting with them and do not have any automated decision support system that helps them when planning activities or detecting any decay of the care-receivers health condition. Furthermore, in terms of group coalitions, while there are some tools that provide this feature, none of them takes into account the complex and fragile environment of the residencies for the elderly and the elderly people needs and likes.

Our proposal amasses these two concepts, producing a tool that aids the decision task of the caregivers and liberates them to perform other tasks, by performing group coalitions of the care-receivers so that they can perform activities; being activity planning one of the most explored domain of the platforms presented.

3. Proposed Model

To make a proper planning of care-receivers allocated per activity during a long period of time (i.e., semester) several parameters should be considered: (i) the physical condition of care-receivers and the requirements of each activity to be performed; (ii) the preferences of care-receivers about activities, to improve their degree of satisfaction; (iii) the friendship relationships of each group.

The most of the nursing homes lack the appropriate number of staff, thus most of the staff is overworked. Another factor to be considered is that the number of activities the nursing homes can offer is limited, so residents have to repeat activities. The use of computational systems that facilitate the activities scheduling process may provide the help that institutions require, streamlining the list of possible activities and groups.

In this section, we describe our proposal for dividing care-receivers into groups to perform activities every day during a period of time using a Coalition Structure Generation Problem. The criterion to generate coalitions relies on physical and psychological aspects of each care-receiver (i.e., preferences, health, friendship, etc.) and the profiles of activities (i.e., physical requirements and the number of people per activity).

The Coalition Structure Generation problem refers to partitioning the components of a set into exhaustive and disjoint coalitions optimizing certain criteria. In our problem, the components of the set are the care-receivers that take part in group activities proposed by a senior residence centre and the criterion to optimize is a social welfare function of each coalition (i.e., the degree of matching between the profile of the care-receivers and the activity in which they participate).

Let $E = \{e_i, ..., e_n\}$ be a set of care-receivers where each individual has a set of features that describes his/her profile. Let $G_j \subseteq E$ be a subset of E called *group*.

The profile of each individual is defined by the following features:

- *Physical status* refers to the physical condition of the individual and can take three values based on his/her medical profile: independent, partially independent, or dependent. Depending on the physical status, there are some activities that are most suitable for an individual. The physical status is known from the beginning and usually remains almost constant during his/her stay in the centre.
- *Preference of activities* refers to how appealing is an activity for an individual. This feature can take three values: appealing, neutral, or non-appealing. We assume that an individual does not have any preference until he/she participates in an activity. Upon the activity completion the feedback is collected about his/her preference.
- *Friendship relationships* represents the social network of the senior residence centre. Nodes represent the individuals and links are weighted bidirectional relations between individuals that take three values: non-friends (i.e., the individuals consider each other annoying), neutral (i.e., the individuals are indifferent with each other), or friends (i.e., the individuals are friends). Initially, information of friendship is not available. After each activity, individuals provide information about his/her relationships with other activity members.
- *Historical activity* the profile stores the sequence of activities already performed by the individual during the planned period. This information is used to avoid the repetition of activities during an specific period of time. Individual preferences, friendship relationships and historical activity profile will be considered in future group activity configurations.

Let $A = \{a_i, ..., a_n\}$ be a set of activities planned for a period of time (i.e., several weeks or months depending on the requirements of the senior residence centre). Each activity is defined by a set of features.

The features associated to an activity are the *type* and the *number of participants*. The activities are classified as psychological or physical. Psychological activities include table games, artistic expression, reading, or religious events, among others. Physical activities include dancing, walks, gardening or culinary lessons, among others. The number of participants is set between a minimum and maximum number of individuals.

We define $< G_j, a_i >$ as a group of individuals that participate in an activity. Given a group $< G_j, a_i >$, the value associated to group is given by a characteristic function $v(<G_j, a_i>) : 2^E xA \rightarrow \mathbb{R}$ that assigns a real-valued payoff to $<G_j, a_i>$. The value of a group $v(<G_j, a_i>)$ is calculated as a linear combination of functions that calculate different types of matching. We define the following functions, whose values ranges from 0 to 1:

- Function $phy(e_j, a_i)$ calculates the degree of match between the physical features of an individual $e_j \in G_j$ and the physical requirements of the activity $a_i \in A$.
- Function $act(e_j, a_i)$ calculates the match between the personal preferences of each individual $e_j \in G_j$ and the activity $a_i \in A$.
- Function $fri(e_j, G_j)$ calculates the degree of friendship of an individual $e_j \in G_j$ with other members of the group $e_k \in G_j : j \neq k$.
- Function $his(e_j, a_i, d)$ penalizes the group if an individual $e_j \in G_j$ has performed the activity $a_i \in A$ in the last d days.

Given the above functions, the value of a group is calculated as:

$$v(G,a) = \sum_{e \in G}(\alpha \cdot phy(e,a) + \beta \cdot act(e,a) \\ + \gamma \cdot fri(e,G) + \delta \cdot hist(e,a)) \tag{1}$$

Note that parameters α, β, γ, and θ are defined to give more relevance to specific features in order to build groups.

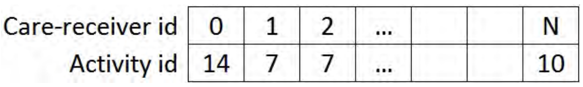
A *group structure* $S = \{<G_1, a_i>, <G_2, a_j>, ..., <G_k, a_n>\}$ is a partition of groups such that $\forall i, j_{i \neq j} : G_i \cap G_j = \emptyset$ and $\cup_{G_i \in S} G_i = E$.

The value of a group structure is denoted by $v(S)$, where $V(S)$ is an evaluation function for the group structure. In this work, we assume that the quality of each group is independent of other groups. Therefore, we can calculate the value of the group structure as:

$$V(S) = \sum_{<G_j, a_n> \in S} v(<G_j, a_n>) \tag{2}$$

The goal is to determine an optimal group structure for the organization of activities $\underset{S \in <2^E xA>}{argmax} \ v(S)$.

It turns out that partitioning a set of elderly individuals into disjoint groups while optimizing a social welfare function corresponds to the formalization of coalition structure generation problems. In order to solve this problem, we propose the use of a genetic algorithm.



Care-receiver id	0	1	2	...			N
Activity id	14	7	7	...			10

Fig. 1. The encoding of a chromosome.

4. Genetic Algorithm Design

Genetic algorithms have been shown to be effective at finding approximate optimal solution, and, in some cases, optimal solutions to combinatorially explosive problems. To solve the coalition formation problem, we proposed a genetic algorithm (see Algorithm 1) that generates successive sets of solutions (generations), where each new generation inherits properties from the best solutions of the previous. Initially, the algorithm creates an initial random population of N individuals. Each individual (chromosome) is a solution to the problem (see Figure 1). Therefore, the size of the chromosome is the number of residents. The chromosome gene order corresponds to the different care-receivers, and gene values correspond to the activity number a care-receiver is engaged. More than one care-receivers engaged in the same activity constitute a group. Each chromosome represents a group structure.

The *fitness function* evaluates the quality of the solutions (i.e., the quality of the individuals). The fitness function in our problem corresponds to function that calculates the value of the group structure (see Equation 3). However, not all the fitness values of the individuals are calculated in the same way. In the described problem, there is a certain type of individuals that must be discarded for future generations, and therefore, they have a 0 fitness value. These individuals are those that are allocated to activities that exceed the maximum number of care-receivers or activities that do not reach the minimum required people.

$$V(S) = \begin{cases} \sum\limits_{<G_j, a_n> \in S} v(G_j) & \text{if } \forall <G_j, a_n> \in S : \\ & min\_size(a_k) < |G_j| < max\_size(a_k) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Genetic operators are applied over the individuals. The algorithm considers four genetic operators (see Figure 2):

- Operator 1 (crossover): Swap two different genes within an individual. This operator allows that two randomly selected participants from different activities swap his/her activities.
- Operator 2 (crossover): Swap all genes with a certain value for all genes with another value within an individual. This operator allows to swap all participants of two activities selected randomly.
- Operator 3 (mutation): Randomly replace all genes with a certain value within an individual with a new, randomly chosen value. This operator allows to randomly change the activity of all the participants of a current coalition. This operator facilitate the inclusion of new activities.
- Operator 4 (mutation): Swap genes with a certain value within an individual with genes with another value within another individual. This operator changes the activity of a group that is part of a planning with the activity of a group in other planning. This operator facilitate the inclusion of new activities.

The *genetic algorithm* is shown in Algorithm 1. A population consisting on a number of possible activities planning is randomly generated. During each iteration (i.e., generation) of the algorithm, a randomly selected genetic operator is applied to each individual (j) of the population and then, its fitness value is calculated according to Equation 3. The resulting individuals after applying the genetic operators (j') are inserted in the new generation. The best N individuals remain in the new generation and the others are removed. The process ends when at least one of these situations occur: (i) the number of generations is exceeded ($k > max\_gen\_k$); (ii) when there are a certain number of generations where there is none individual in the new generation that has a higher value of fitness than the best individual in previous generations ($q > max\_gen\_q$); (iii) when the algorithm exceeds the time limit ($t_{end} - t_{start} > T$).

5. Experiments

In this section, we show different experiments that were carried out in order to evaluate the efficiency and the performance of the algorithm proposed. First, we compared the genetic algorithm with another algorithm that is based on linear programming. Then, we compared the results of the genetic algorithm depending on the relevance of the parameters of the fitness function.

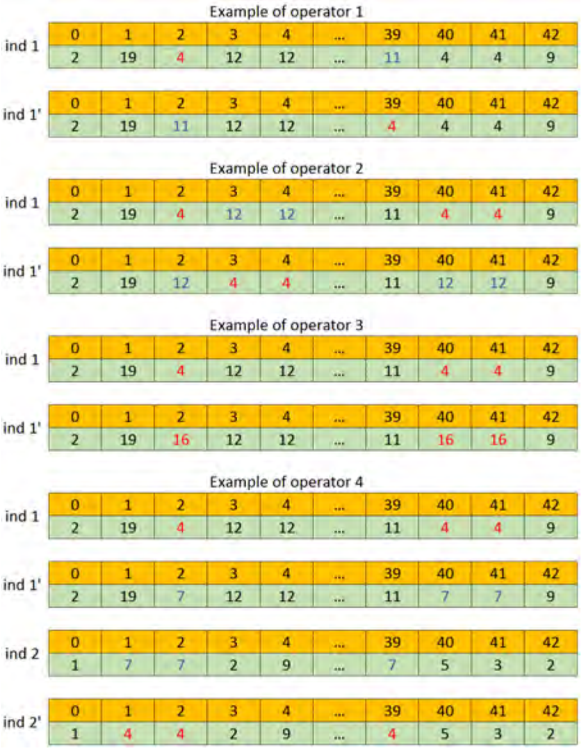
Algorithm 1 The Evolution Algorithm

```
Generate an initial population of N random individuals
Evaluate the fitness of each individual of the population N
Select the best solution s
Number of generations k = 0
Number of generations without improving the solution q = 0
Max number of generations without improving the solution $max\_gen\_q$
Max number of generations $max\_gen\_k$
Temporal constraints $t_{start} = System\_time$; $t_{end} = 0$; $T_{limit} = T$
while ($k \leqslant max\_gen\_k \wedge q \leqslant max\_gen\_q \wedge t_{end} - t_{start} \leqslant T$) do
    for ($j = 0; j < N; j++$) do
        randomly apply one of the genetic operators over individual j
        evaluate the fitness value of j and j' (i.e., j' is the resulting individual after applying an operator on chromosome j)
        insert j and j' in the new generation
    end for
    selection of N best individuals
    selection of the best individual s'
    if (V(s') ⩽ V(s)) then
        q++
    end if
    k++
    $t_{end} = System\_time$
end while
```

Finally, we tested the efficiency of two additional fitness functions.

For each set of experiments, groups of people were formed during 30 days in which the value of the group structure was calculated for each day, considering that each person could carry out a single activity per day, and a penalization was introduced if the same activity was repeated in a three-days period. The size of each group ranged from 3 to 5 people per group. It must be pointed that some activities could be carried out by different group sizes while others must be only carried out by a specific number of group size. Note that an activity might have no persons.

Regarding the configuration of the genetic algorithm, we considered a population of $N = 1000$ individuals per iteration. After the application of the genetic operators, we obtain a population of 2000 indi-



Example of operator 1

	0	1	2	3	4	...	39	40	41	42
ind 1	2	19	4	12	12	...	11	4	4	9

	0	1	2	3	4	...	39	40	41	42
ind 1'	2	19	11	12	12	...	4	4	4	9

Example of operator 2

	0	1	2	3	4	...	39	40	41	42
ind 1	2	19	4	12	12	...	11	4	4	9

	0	1	2	3	4	...	39	40	41	42
ind 1'	2	19	12	4	4	...	11	12	12	9

Example of operator 3

	0	1	2	3	4	...	39	40	41	42
ind 1	2	19	4	12	12	...	11	4	4	9

	0	1	2	3	4	...	39	40	41	42
ind 1'	2	19	16	12	12	...	11	16	16	9

Example of operator 4

	0	1	2	3	4	...	39	40	41	42
ind 1	2	19	4	12	12	...	11	4	4	9

	0	1	2	3	4	...	39	40	41	42
ind 1'	2	19	7	12	12	...	11	7	7	9

	0	1	2	3	4	...	39	40	41	42
ind 2	1	7	7	2	9	...	7	5	3	2

	0	1	2	3	4	...	39	40	41	42
ind 2'	1	4	4	2	9	...	4	5	3	2

Fig. 2. Examples of genetic operators.

viduals. The best 1000 individuals will be selected for the next iteration.

We defined different scenarios in order to consider different population of people and activities. Our starting scenario consists of 43 people and 20 possible activities. We combined heterogeneous populations in which some people had a high friendship degree while others had a low friendship degree. Similarly, some activities were preferred by a high number of people while others were only preferred by few people. These proposed scenarios were configured as follows:

- *Scenario 1*: 43 persons and 20 possible activities to carry out.
- *Scenario 2*: A selection of the 20 persons with the highest degree of friendship and 20 possible activities to carry out.
- *Scenario 3*: A selection of the 20 persons with the lowest degree of friendship and 20 20 possible activities to carry out.
- *Scenario 4*: 43 persons and a selection of the 10 most preferred activities.
- *Scenario 5*: 43 persons and a selection the 10 least preferred activities.

Physical status of e_j	**Physical requirements of a_i**	**value**
Independent	No requirements	1
Independent	Require good physical status	1
Partially independent	No requirements	1
Partially independent	Require good physical status	0.5
Dependent	No requirements	1
Dependent	Require good physical status	0

Table 1

Values for $phy(e_j, a_i)$

Preference of e_j about a_i	**value**
Appealing	1
Neutral	0.5
Non-appealing	0

Table 2

Values for $act(e_j, a_i)$

Degree of friendship between e_j about e_k	**value**
Friends	1
Neutral	0.5
Non-friends	0

Table 3

Values for $fri(e_j, G_j)$

e_j has performed a_i in the last d days	**value**
Yes	0
No	1

Table 4

Values for $his(e_j, a_i, d)$

Each individual execution was repeated 10 times for each scenario and Student's t-tests was performed to assess whether the differences among the strategies were significant. In terms of hardware, experiments were launched in a virtual machine with 6 processor Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70GHz with 4GB of RAM.

In the Tables 1, 2, 3, and 4 it is shown how the values for the functions defined in Section 3 are calculated for the following experiments. It must be pointed out that function $fri(e_j, G_j)$ considers the average value of friendship for each pair of individuals according to what is shown in Table 3.

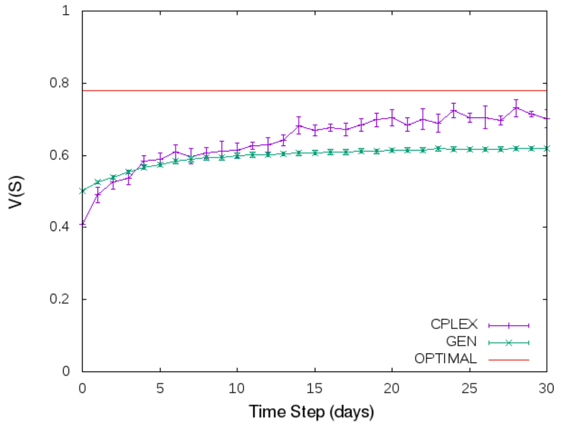


Fig. 3. Scenario 1.

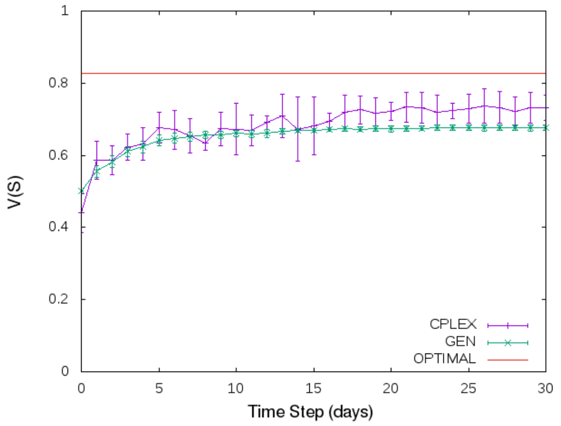


Fig. 4. Scenario 2.

5.1. *Genetic algorithm vs Linear programming*

The first set of experiments was aimed at testing the efficiency and performance of the genetic algorithm proposed. For each scenario, the value of each group is calculated considering that each factor (physical condition, preferences, friendship, and previous activities performed) of the fitness function has the same weight according to the Equation 3.

In order to compare the efficiency of the genetic algorithm proposed, we also applied the commercial software *ILOG CPLEX 12.5* [1]. This software solves the problem as a linear programming problem [31] obtaining the best configuration for each day. In order to compare both algorithms, the value of the group structure is obtained according to Eq. 2. In this algorithm, the best solution can be found since the whole range of solutions is explored. In contrast, the computational time required to find the best configuration is expected to be high.

Figure 3 shows the results of the *Scenario 1*. This figure shows the value of the group structure of each algorithm with the 95% confidence interval. In addition, the upper bound of the highest value of the group structure is also represented as a continuous line above all the strategies. This upper bound represent an scenario in which all the preferences are known and any penalization is carried out. This bound can be obtained once every participant has defined his/her preference for each activity and his/her friendship relationship with any other participant.

As it can be observed in the figure, as the more information was considered for group formation, the higher the value of the group structure was. Thus, both the *genetic algorithm* and *CPLEX* improved the performance during the 30 days, getting closer to the optimal. Although differences among both strategies were significant from day 15 on, the differences between the two strategies for all days that is lower than 0.1 in day 30, showing that the performance of the *genetic algorithm* algorithm is quite close to the *CPLEX*.

However, computational differences were notable. While the time required to obtain the optimal coalitional structure for each of the 30 days by the *genetic algorithm* was $14.21s \pm 0.41$, the time required by *CPLEX* was $689.56s \pm 47.78$. These differences are explained since *CPLEX* explores all the solutions.

In Figure 4 we can observe the results of the *Scenario 2*. Similar to the previous scenario, the performance of both strategies increased during the 30 days as more information was considered. In contrast, the differences between the *genetic algorithm* and *CPLEX* were reduced and the average values in the day 30 were lower than 0.06. Regarding computational time, since the population was lower than in *Scenario 1*, the time required by both strategies was also lower. Despite this, the *genetic algorithm* performance was much better, requiring $0.68s \pm 0.01$ to execute and iteration, while *CPLEX* required $3.76s \pm 0.12$.

Figure 5 shows the performance of *Scenario 3*, which was similar to the previous scenario. Since this correponds to a configuration in which persons had a low degree of friendships, the values were low, and therefore, differences between both strategies were

---

[1] http://www.ibm.com/software/commerce/optimization/cplex-optimizer/ - Last access: 25/07/2016

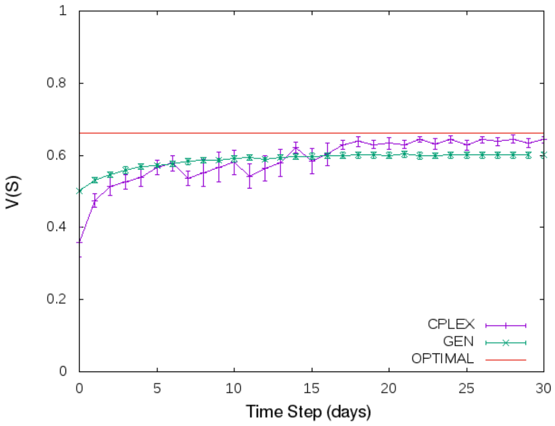


Fig. 5. Scenario 3.

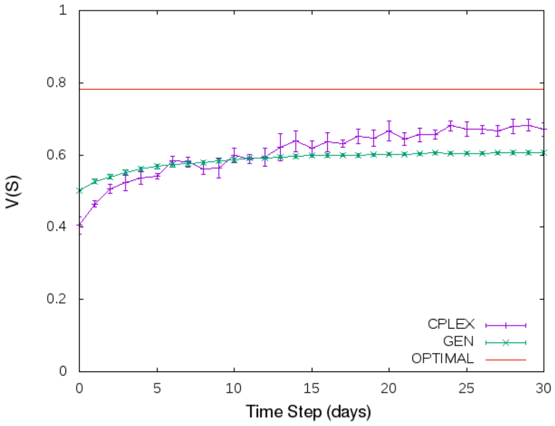


Fig. 6. Scenario 4.

also few. Although these were significant from day 17 on, the difference between the average values of the coalitional structure obtained in day 30 by the *genetic algorithm* and *CPLEX* were lower than 0.04. In this scenario, computational consumption was similar to the *Scenario 2*, being $0.72s \pm 0.02$ for the *genetic algorithm* and $3.84s \pm 0.18$ for *CPLEX*.

The next scenario, considered the same population that *Scenario 1* but the half of the activities (Figure 6). Similar to the previous scenarios, in *Scenario 4*, the performance of both strategies increased during the 30 days. In this case, the differences between *CPLEX* and *genetic algorithm* were higher than in other scenarios, becoming significant from day 16 on and achieving a difference of 0.06 between both strategies in day 30. This can be explained due to the fact that $v(S)$ achieves higher values than in previous scenarios. Since the population was lower than in *Scenario 1*, the computational time required by the *genetic algorithm* was lower, being $1.74s \pm 0.05$. In contrast, this is not relevant for *CPLEX*, whose computational time was $692.10s \pm 34.60$

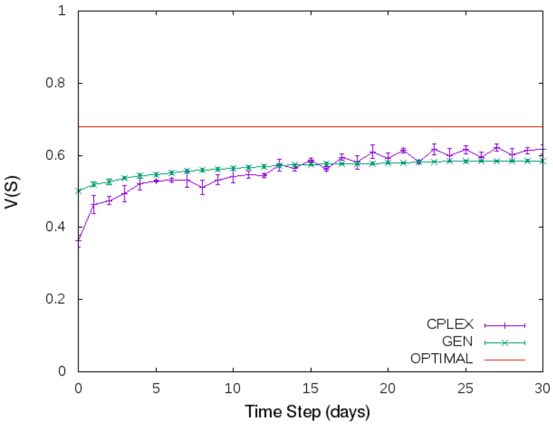


Fig. 7. Scenario 5.

Strategy	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Genetic algorithm	14.21± 0.41	0.68± 0.01	0.72± 0.02	1.74 ± 0.05	1.92 ± 0.06
CPLEX	689.56± 47.78	3.76± 0.12	3.84± 0.18	692.10± 34.60	686.80± 31.82

Table 5
Time consumption (in seconds).

Finally, Figure 7 shows the performance of the *Scenario 5* for the *genetic algorithm* and *CPLEX*. This scenario is similar to the previous one but the activities considered were those preferred by the lowest number of people. This caused that the values of the coalitional structures were lower compared to *Scenario 4*. In this case, the differences between both strategies in day 30 were around 0.03. Computational times were similar to the previous scenario, being $1.92s \pm 0.06$ for the *genetic algorithm* and $686.80s \pm 31.82$ for the CPLEX.

As a general conclusion, *CPLEX* obtains the optimal coalitional structure according to the values of physical condition, preferences, and friendships known at each moment. It is also observed that the performance of the *genetic algorithm* was quite close to the *CPLEX* performance, even not all possibilities are explored. However, the average time required for obtaining the solutions were considerably different, requiring much more time for *CPLEX* as we can observe in Table 5.1. In addition, as it can be appreciated, as more complex scenarios are considered, more computational time is required, which would make some problems to become unmanageable at a reasonable time. In contrast, since the *genetic algorithm* provides quite optimal solutions in a response time much more lower, much complex problems could be managed.

5.2. Relevance of parameters of the fitness function

Following, we present some experiments focused on testing the flexibility of the model by adapting the weight of the parameters of the fitness function defined in Equation 1. These experiments tested how we could adapt the fitness function according to the requirements of the context.

Note that the original fitness function gave the same weight to all the parameters. To measure the influence of these parameters, we defined two new fitness functions that modified some of these parameters. One function gave more relevance to the preference of activities than the other parameters (labeled as *ACT*) and the other gave more relevance to the friendship relationships (labeled as *FRI*). We tested these two fitness function on those scenarios that changed both the available activities and the population, i.e. Scenarios 2 to 5.

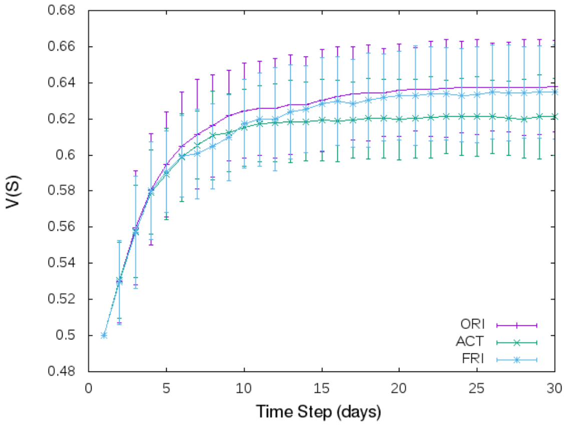


Fig. 8. Scenario 2. Changing the weight of the parameters.

Figure 8 compares the performance of the original fitness function (labeled as *ORI*) with these new fitness functions on the Scenario 2. As it can be observed, the performance of the *ACT* function was lower than the other two functions from day 15 on, while the performance of *ORI* and *FRI* was quite similar. This can be explained due to the fact that this scenario considered the 20 persons with the highest degree of friendship. Therefore, if the parameter related to the friendship relationship was given more importance than the rest, groups with high degree of friendship were expected to be part of the solutions. In contrast, the *ACT* function might not found groups with high degree of friendship since the parameter that was given more importance was the preference of activities.

Differences between *ACT* and *ORI* were significant from day 15 on, while differences between *ACT* and *FRI* were significant from day 20 on. It can be noted that the *ORI* function was slightly better than the *FRI* function, although their differences were not significant. This can be caused since *FRI* gave a high weight to the degree of friendship, but the friendship degree among the population of persons was not enough to obtain better solutions than using the *ORI* function.

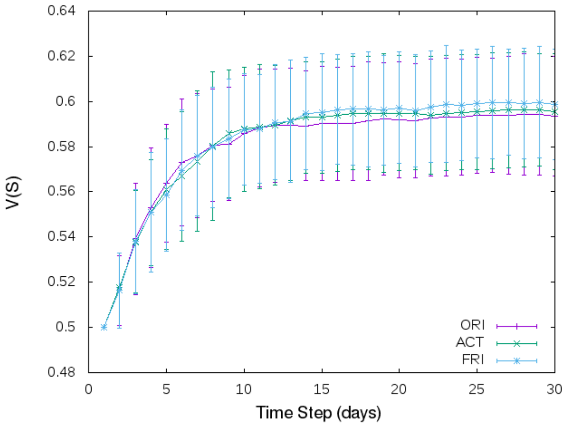


Fig. 9. Scenario 3. Changing the weight of the parameters.

The above behaviour changed on the Scenario 3, which is shown in Figure 9. In this case, the function with the worse performance was the *ORI* function, while the differences between *FRI* and *ACT* were reduced. This was caused because of the function that was focused on finding groups with high degree of relationships, was not able to find good solutions since the persons considered in this scenario were those with the lowest degree of relationships. In this scenario, differences among the functions were not significant.

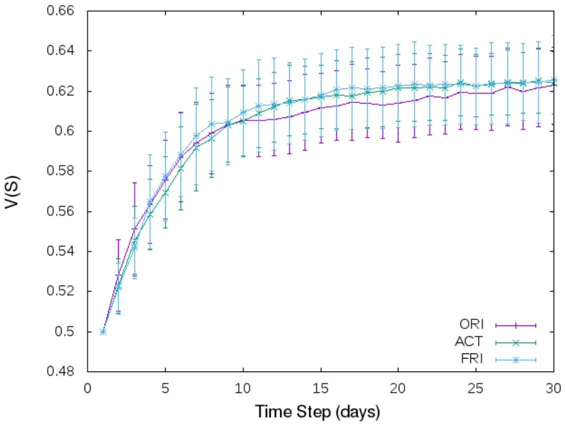


Fig. 10. Scenario 4. Changing the weight of the parameters.

The next experiment applied the three functions on the Scenario 4 and the results are shown in Figure 10. As it can be observed, the performance of the *ORI* function was worse than the other two functions, but differences were not significant. It can be concluded that considering the whole population with the 10 most preferred, that fact of grouping people focusing on their activities preference or focused on their friendship degree had not any significant difference.

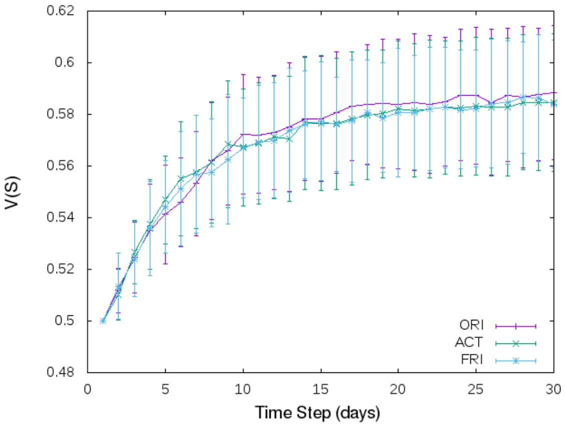


Fig. 11. Scenario 5. Changing the weight of the parameters.

Finally, Figure 11 shows the performance of the three functions on the Scenario 5. In this case, the *ORI* function performed slightly better than the other two functions because of this scenario considered the 10 less preferred activities. However, similar to the previous experiment, there were not significant differences under these conditions.

In summary, throughout these experiments we could test how the parameters of the fitness function can be adjusted depending on which aspects we want to prioritize in the solutions found. The objective of these experiments was not to find the best solution for each scenario but testing how we could change the weights of the parameters in order to focus the solutions. Note that depending on the specific characteristics of specific populations and activities, we could try to find the most appropriated combination of parameters that maximize the value of the solutions that are found. In addition, new parameters could be added depending on the circumstances of our context.

5.3. *Changing the fitness function*

The following set of experiments was aimed at testing the flexibility of the model for including new fitness functions. As we observed in previous experiments, the parameters of the fitness function can be adjusted in order to give more relevance to those aspects that we consider more critical in our context. However, there might be situations in which these adjustments are not enough for our requirements. Note that the fitness function of Equation 2 is oriented to maximize the value of the group structure, however, nothing is considered at groups level. Thus, the solution obtained by the algorithm may provide a group structure in which the differences among the values of the groups is high (i.e. some groups have a high value while others have a low value). Hence, this approach may not be suitable for scenarios in which a well-balanced distribution of groups is required.

To deal with this issue, following we propose two additional fitness functions. The first of these functions calculates the value of the group structure as the product of the value of the groups instead of the sum:

$$v(S) = \prod_{<G_j,a_n>\in S} v(<G_j,a_n>) \tag{4}$$

In this case, solutions composed by groups with similar values may have a higher global value of the group structure than solutions in which some of the groups have very low values. From now on, we will refer to the original fitness function as *F1* and this new function as *F2*.

The other fitness function proposed (from now on, refered as *F3*) is oriented to penalize solutions with large differences between the group with the highest value and the group with the lowest value. In this case, we include this factor as a weigh applied to the original fitness function:

$$v(S) = \sum_{<G_j,a_n>\in S} v(<G_j,a_n>) \times (1-(max(S)-min(S))) \quad (5)$$

where:

$$max(S) = max\ v(<G_i,a_n>) \ \ / \ \ <G_i,a_n>\in S$$
$$min(S) = min\ v(<G_i,a_n>) \ \ / \ \ <G_i,a_n>\in S$$

Considering the original function and the two new functions defined above, following we show their behaviour on the Scenarios 2 to 5.

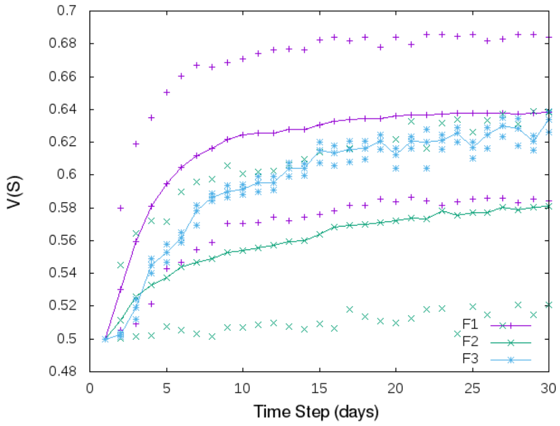


Fig. 12. Scenario 2. Performance of the three fitness functions.

Figure 12 shows the $v(S)$ value of the group structure for the three fitness functions applied to the Scenario 2, together with the highest and the lowest values of the groups that compose the solution. It can be observed that the performance of *F1* was greater than *F2* and *F3*, and differences were significant among the three functions. However, the results shown by *F3* were groups whose value was very similar, i.e. the groups were well-balanced. As it can be appreciated, the average value, the highest, and the lowest were all of them very similar for *F3*, while these values were much more different in *F1* and *F2*.

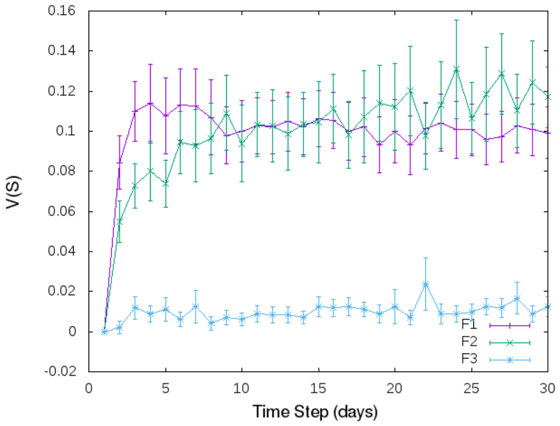


Fig. 13. Scenario 2. Differences between the groups with the highest and lowest values.

This can be observed in more detail in Figure 13, which shows the average difference between the groups with highest and lowest values that are part of the solutions. As it can be appreciated, differences among the group with the highest value and the group with the lowest value were around 10 times higher for *F1* and *F2*. In more detail, these ranged from 0.09 and 0.12 for *F1* and *F2*, while these differences ranged from 0.006 and 0.012 for *F3*. As a general conclusion, *F1* provided groups whose average value was higher than the other functions, but differences among the groups were larger than *F3*. In contrast, *F3* provided solutions whose groups were much more well-balanced. In this scenario, *F2* had the worst performance.

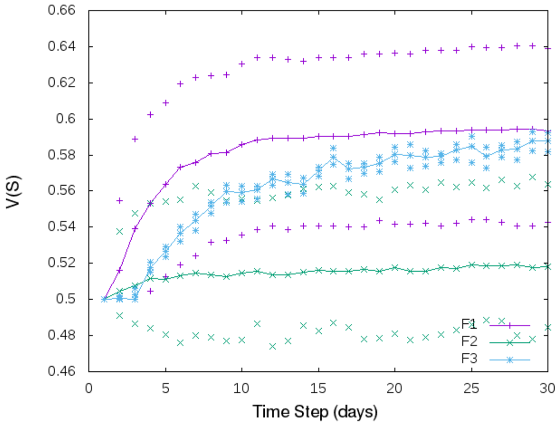


Fig. 14. Scenario 3. Performance of the three fitness functions.

The above results were very similar under the conditions of the Scenario 3. Figure 14 shows the performance of the three functions on this scenario. As it can be observed, the behaviour of the three functions was similar to the previous scenario. Function *F1* provided the better results in terms of average values. However, differences between *F1* and *F3* were getting lower each iteration. Thus, their differences were not significant in the last two days. The behaviour of *F2* was again lower than the other two functions. What is more, in this scenario that considered the 20 persons with the lowest degree of friendship, the evolution of *F2* was very poor along the 30 days. As it can be appreciated, the average value was 0.514 in day 2 and the maximum average value reached during the 30 days was 0.529. This can be explained due to the fact that in this scenario, in which there are some values close to 0, the function that considers a product is very influenced by these values. In contrast, the other functions can balance the appearance of values close to 0.

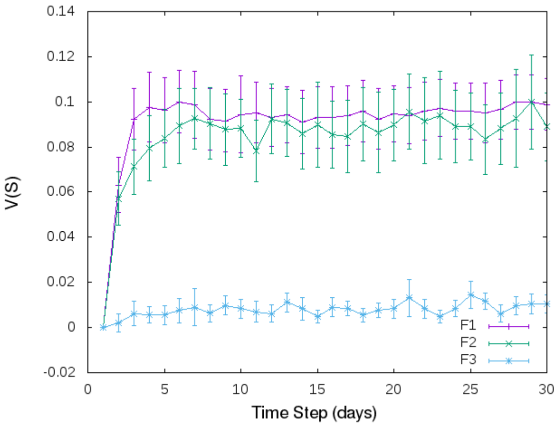


Fig. 15. Scenario 3. Differences between the groups with the highest and lowest values.

Similar to the previous experiment, *F3* provided solutions whose groups were much more well-balanced. Figure 15 shows the difference between the group with the highest value and the group with the lowest value for the three functions. In this case, these differences ranged from 0.07 and 0.10 for *F1* and *F2* and between 0.006 and 0.01 for *F3*.

Figure 16 shows the behaviour of the three functions on Scenario 4. In contrast to the behaviour in Scenarios 2 and 3, in this scenario the average performance of *F1* and *F2* was very similar, showing that differences between both functions were not significant. This can be explained due to the fact that the number of groups created in this scenario is 9 or 10, while the previous scenarios only required 4 or 5 groups. Thus, the number of groups involved in the fitness function is larger and this may cause that the performance of *F2*, which consider a product of values, was much more similar to *F1*.

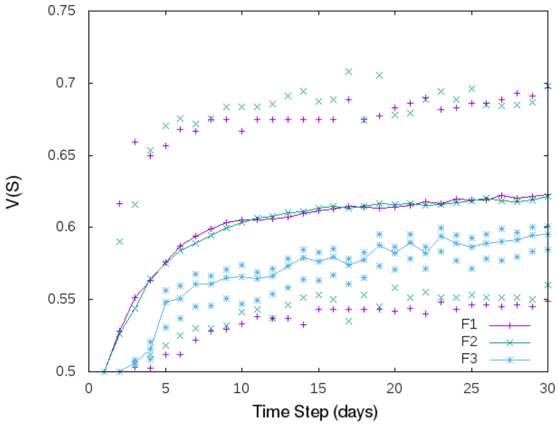


Fig. 16. Scenario 4. Performance of the three fitness functions.

In addition, both the maximum and minimum values were higher in *F2* than in *F1*. This could be caused because of higher values of groups are more relevant in a product. In contrast, the behaviour of *F3* was different from the other two functions. First, the majority of the average and maximum values were lower in *F3* than in *F1* and *F2*, but the minimum values were usually higher. This caused that solutions found by *F3* considered groups whose values were very similar.

As it can be appreciated in Figure 17, the values of *F1* and *F2* ranged from 0.11 to 0.16, while the values of *F3* were 10 times lower, between 0.01 and 0.02.

Following, we show the results of the three functions on the Scenario 5 in Figures 18 and 19. The behaviour of these functions were quite similar to the previous scenario, but two differences can be appreciated. First, comparing *F1* and *F2* it can be observed that the groups with highest values of *F1* were usually higher than *F2* and the groups with the lowest values of *F1* were usually lower than *F2*. As it is shown in Figure 19, this caused that differences between the groups with the highest and the lowest value were higher in *F1* than in *F2*. Therefore, the solutions provided by *F1* were composed by groups with larger differences (around 0.16 and 0.19) than the solutions found by

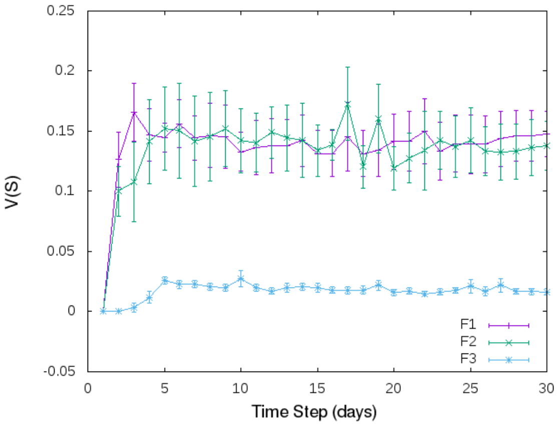


Fig. 17. Scenario 4. Differences between the groups with the highest and lowest values.

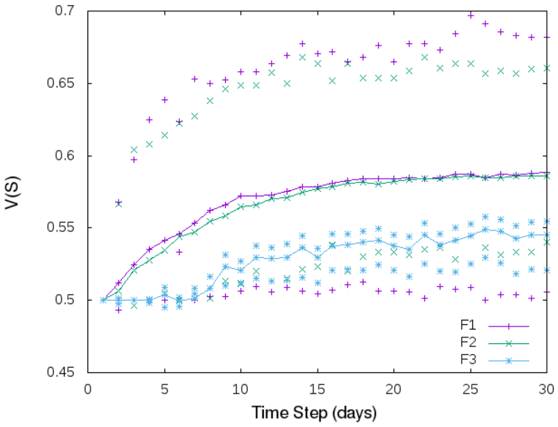


Fig. 18. Scenario 5. Performance of the three fitness functions.

F2, which were more well-balanced (around 0.11 and 0.13). As in the previous experiments, the most well-balanced groups were found by *F3*, whose differences usually ranged between 0.01 and 0.03).

The second difference that can be appreciated regarding the Scenario 4 is that the minimum value of *F3* was not clearly higher than *F2*. These two differences can be due to the fact that in Scenario 5 the values of the group structure of the solutions were lower since this scenario considered only the 10 activities less preferred. Thus, the behaviour of the functions were slightly different.

As a general conclusion, it can be stated that *F3* should be more suitable for scenarios in which finding groups with similar values is more important, while *F1* and *F2* should usually perform better to find groups whose average value is higher. Apart from this, what is more important is that the model proposed provides enough flexibility to adjust not only the parameters but also to modify the fitness function depending on the requirements of our context. In the above experiments, we considered different scenarios with specific characteristics, but our model could be extended to be applied to other scenarios.

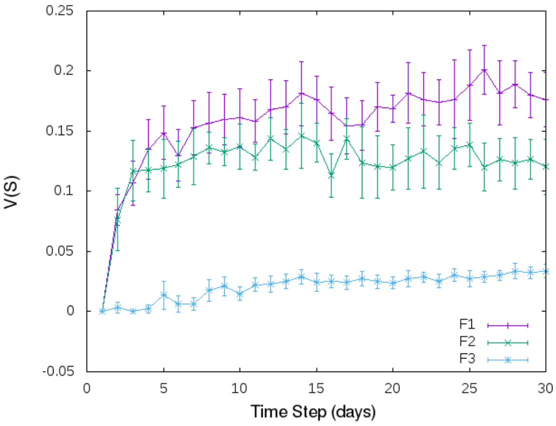


Fig. 19. Scenario 5. Differences between the groups with the highest and lowest values.

6. Conclusions

In this paper, we presented a model for dividing care receivers into groups to perform activities with elderly people. This model allows the representation of physical requirements of the individuals but also preferences and social relationships. The model is also able to learn these features as activities are performed by care receivers and improves next activities configurations.

We represented the problem of finding the most suitable grouping as a Coalition Structure Generation problem, which we solved by implementing a Genetic Algorithm. The set of experiments evaluates the performance and the flexibility of the proposed model. Firstly, we evaluated the solution obtained by our algorithm and the optimal solution obtained by *CPLEX*. The results show that the solution provided by our algorithm was really close to the optimal values for all the scenarios proposed. What is more, the computational time required to find the solution was really small compared to the time required by *CPLEX*, which explores all the solutions. Therefore, our algorithm could be applied in more complex problems with large populations and activities. Secondly, we tested the genetic algorithm in scenarios where could be interesting prioritze in certain aspects of the fitness function. The results show that although the weights of the parameters of the fitness function are not balance, the algorithm is able to find an appropriate solution in different scenarios. Finally, we tested a set of fitness functions to generate equally distributed coalitions inside the coalition structure. In general, the results show that the proposed model based on the genetic algorithm is able to offer a good performance and to provide a solution close to the optimal. Moreover, it provides flexibility to deal with different scenarios where weights of parameters and fitness functions could be adjusted to offer more balance solutions.

As future work, we are considering the inclusion of other parameters in the fitness function as well as other operators and search strategies. It must be pointed out that our model considers that preferences about activities and friendship degrees are similar during the execution. However, in a more realistic scenario, these can be dynamic. Therefore, as one of the points for future works, we plan to extend our model in order to include this dynamicity. We plan to test the performance of the algorithm and the suitability of the solution provided by the algorithm in a real context. For scenarios where the number of activities, the number of participants an the activities size increases, we plan to parallelize the algorithm.

Acknowledgment

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013. A. Costa thanks the FCT – Fundação para a Ciência e Tecnologia the Post-Doc scholarship with the Ref. SFRH/BPD/102696/2014. This work is also partially supported by the MINECO/ FEDER TIN2017-89156-R and the Post-Doc scholarship with the Ref. SP20170057.

References

[1] J.M. Alberola, E. del Val, V. Sánchez-Anguix, A. Palomares and M.D. Teruel, An artificial intelligence tool for heterogeneous team formation in the classroom, *Knowledge-Based Systems* (2016).

[2] J.M. Alberola, E. del Val, V. Sánchez-Anguix and V. Julian, A general framework for testing different student team formation strategies, in: *Methodologies and Intelligent Systems for Technology Enhanced Learning, 6th International Conference*, AISC Springer Verlag, 2016.

[3] J.M. Alberola, E. del Val, V. Sanchez-Anguix and V. Julian, Simulating a collective intelligence approach to student team formation, in: *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2013, pp. 161–170.

[4] S. Whittier, D.W. Coon and J. Aaker, Caregiver Support Interventions, 2002.

[5] J. Cohen-Mansfield, K. Thein, M. Dakheel-Ali and M.S. Marx, Engaging nursing home residents with dementia in activities: The effects of modeling, presentation order, time of day, and setting characteristics, *Aging & Mental Health* **14**(4) (2010), 471–480. doi:10.1080/13607860903586102.

[6] M. Guerin, S. Mackintosh and C. Fryer, Exercise class participation among residents in low-level residential aged care could be enhanced: a qualitative study, *Australian Journal of Physiotherapy* **54**(2) (2008), 111–117, ISSN 0004-9514.

[7] S.L. Logan, B.H. Gottlieb, S.B. Maitland, D. Meegan and L.L. Spriet, The Physical Activity Scale for the Elderly (PASE) Questionnaire; Does It Predict Physical Health?, *International Journal of Environmental Research and Public Health* **10**(9) (2013), 3967, ISSN 1660-4601.

[8] P. Chang, L. Wray and Y. Lin, Social relationships, leisure activity, and health in older adults, *Health Psychol* **33**(46) (2014), 516–523.

[9] H.S. Minhat and R. Mohd Amin, Sociodemographic Determinants of Leisure Participation Among Elderly in Malaysia, *Journal of Community Health* **37**(4) (2011), 840–847.

[10] M.C. Ashe, W.C. Miller, J.J. Eng and L. Noreau, Older Adults, Chronic Disease and Leisure-Time Physical Activity, *Gerontology* **55**(1) (2009), 64–72.

[11] S.K. Bal, Leisure activity and risk of dementia scrabble, anyone?, *Canadian Family Physicia* **50** (2004), 51–53.

[12] C. Haslam, S. Alexander Haslam, C. Knight, I. Gleibs, R. Ysseldyk and L.-G. McCloskey, We can work it out: Group decision-making builds social identity and enhances the cognitive performance of care residents, *British Journal of Psychology* **105**(1) (2014), 17–34, ISSN 2044-8295.

[13] S. Freeman, L. Spirgiene, M. Martin-Khan and J.P. Hirdes, Relationship between restraint use, engagement in social activity, and decline in cognitive status among residents newly admitted to long-term care facilities, *Geriatrics & Gerontology International* (2016), ISSN 1447-0594.

[14] N.J. Stumbo and C.A. Peterson, *Therapeutic Recreation Program Design: Principles and Procedures*, Pearson/Benjamin-Cummings, 2009. ISBN 9780321541888.

[15] Â. Costa, J.C. Castillo, P. Novais, A. Fernández-Caballero and R. Simoes, Sensor-driven agenda for intelligent home care of the elderly, *Expert Systems with Applications* **39**(15) (2012), 12192–12204, ISSN 0957-4174. doi:10.1016/j.eswa.2012.04.058. http://linkinghub.elsevier.com/retrieve/pii/S0957417412006550.

[16] A. Costa, P. Novais and R. Simoes, A caregiver support platform within the scope of an ambient assisted living ecosystem., *Sensors (Basel, Switzerland)* **14**(3) (2014), 5654–5676, ISSN 1424-8220. doi:10.3390/s140305654.

[17] J. Yamamoto and K. Sycara, A stable and efficient buyer coalition formation scheme for e-marketplaces, in: *Proceedings of the fifth international conference on Autonomous agents*, ACM, 2001, pp. 576–583.

[18] M. Tsvetovat, K. Sycara, Y. Chen and J. Ying, Customer coalitions in electronic markets, in: *International Workshop on Agent-Mediated Electronic Commerce*, Springer, 2000, pp. 121–138.

[19] T. Ito, H. Ochi and T. Shintani, A group-buy protocol based on coalition formation for agent-mediated e-commerce, *IJCIS* **3**(1) (2002), 11–20.

[20] B. Blankenburg and M. Klusch, On safe kernel stable coalition forming among agents, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, IEEE Computer Society, 2004, pp. 580–587.

[21] N. Ohta, V. Conitzer, Y. Satoh, A. Iwasaki and M. Yokoo, Anonymity-proof shapley value: extending shapley value for coalitional games in open environments, in: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 927–934.

[22] Y. Zick and E. Elkind, Arbitrators in overlapping coalition formation games, in: *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 55–62.

[23] T. Rahwan, T.P. Michalak, M. Wooldridge and N.R. Jennings, Coalition structure generation: A survey, *Artificial Intelligence* **229** (2015), 139–174.

[24] E. Rohn, M. Kalech and A. Diskin, Coalition Formation Decision Support System, *Social Science Computer Review* **34**(4) (2016), 437–455.

[25] F. Bistaffa, A. Farinelli and S.D. Ramchurn, Sharing rides with friends: a coalition formation algorithm for ridesharing, in: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[26] C. Ding, F. Xia, G. Gopalakrishnan, W. Qian and A. Zhou, TeamGen: An Interactive Team Formation System Based on Professional Social Network, in: *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 195–199. ISBN 978-1-4503-4914-7.

[27] S.M. Hosseini, P. Akhavan and M. Abbasi, A knowledge sharing approach for RD project team formation, *VINE Journal of Information and Knowledge Management Systems* **47**(2) (2017).

[28] J.M. Alberola, E. Del Val, V. Sanchez-Anguix, A. Palomares and M.D. Teruel, An artificial intelligence tool for heterogeneous team formation in the classroom, *Knowledge-Based Systems* **101** (2016), 1–14.

[29] S. Antunes Silva, S. Renato Dias and L. Soriano Marcolino, Bang: a system for training and visualization in multi-agent team formation, in: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 1781–1783.

[30] I.E. Allen and C.A. Seaman, Likert scales and data analyses, *Quality progress* **40**(7) (2007), 64.

[31] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki and M. Yokoo, Coalition Structure Generation Utilizing Compact Characteristic Function Representations, in: *Principles and Practice of Constraint Programming - CP 2009*, Vol. 5732, Springer, 2009, pp. 623–638. ISBN 978-3-642-04243-0.

[32] J. Yamamoto and K. Sycara, A stable and efficient buyer coalition formation scheme for e-marketplaces, in: *Proceedings of the fifth international conference on Autonomous agents*, ACM, 2001, pp. 576–583.