Available online at www.sciencedirect.com
SCIENCE DIRECT
Engineering Applications of Artificial Intelligence 19 (2006) 569–578
Engineering Applications of ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Solving terminal assignment problems with groups encoding: The wedding banquet problem☆

Sancho Salcedo-Sanz*, José A. Portilla-Figueras, Fernando García-Vázquez, Silvia Jiménez-Fernández

Department of Signal Theory and Communications, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain

Received 11 February 2005; received in revised form 6 September 2005; accepted 30 October 2005
Available online 18 January 2006

Abstract

This paper presents an extension of the terminal assignment problem (TA) in the case that groups of terminals must be assigned together. We analyze this situation by means of an equivalent problem: the wedding banquet problem (WBP). We provide a description of the problem and its mathematical definition. We also describe an application of the WBP to mobile communications network design. Two hybrid metaheuristics algorithms for the WBP are presented in the paper. We test their performance in several computational experiments, including synthetic instances of the WBP, and a mobile network design application.

Keywords: Terminal assignment problem; Wedding banquet problem; Hybrid metaheuristics

1. Introduction

An important class of combinatorial optimization problems with constraints (Smith et al., 1996; Smith, 1999) can be defined as the maximization or minimization of a goal function (usually in a binary search space) subject to a set of constraints, which have to be satisfied for a solution to be feasible. Some of these problems arise frequently in computational engineering and computer science fields, such as scheduling problems (Wang and Ansari, 1997; Salcedo-Sanz et al., 2003), assignment problems (Menon and Gupta, 2004) or problems of systems design (Soni et al., 2004). One of these combinatorial optimization problems with a wide range of applications is the so-called *terminal assignment problem* (TA hereafter), which in the last few years has been tackled using several different heuristics approaches, (Abuali et al., 1994; Khuri and Chiu, 1997; Salcedo-Sanz and Yao, 2004; Yao et al., 2004).

In this paper, we investigate an extension to the TA, which includes groups of terminals that must be assigned together to the same concentrator. This could have interesting applications in mobile telecommunications networks design (Kershenbaum, 1993; Chu et al., 2000; Soni et al., 2004), where grouping of several network elements (base transceivers stations (BTSs) for example) must be grouped to improve the reliability of the network and to reduce the investment costs. To carry on an analysis of the extension from TA to TA with groups of terminals, we propose an equivalent problem, that we call the *wedding banquet problem* (WBP).

It is amazing that even in a social event like a wedding, the couple who is going to get married often have to solve a combinatorial optimization problem. In fact, they have to solve a kind of TA with groups of guests serving as terminals, and tables serving as concentrators: In the majority of weddings, there is a banquet after the ceremony, in which family and friends of the bride and the groom celebrate together the great event. The problem arises when the couple has to decide the guests who must

☆This work has been partially supported by Universidad de Alcalá, through Projects UAH PI2005/078 and UAH PI2005/079, and by the NoE Euro NGI through Project IST 50/7613 of the VI European Framework Programme.
*Corresponding author. Tel.: +34 91 885 6731; fax: +34 91 624 8749.
E-mail address: sancho.salcedo@uah.es (S. Salcedo-Sanz).

0952-1976/$ - see front matter
doi:10.1016/j.engappai.2005.10.003

share table in the banquet. Of course, there is a maximum number of guests in each table (the capacity constraint of the TA), and there are groups of guest that must be seated together (families or very close friends). The objective function of the WBP is to maximize the *happiness* in the banquet, assigning together as many people who get on well as possible.

In this paper, we provide an analysis of the WBP. This includes to describe the model that we consider and relate it with the TA with groups of terminals (Brudaru, 2003; Brown and Sumichrast, 2005). Also, we show its applications to mobile network design problems. Finally, we present two hybrid metaheuristic approaches to solve the WBP, since traditional optimization approaches are not applicable to this kind of combinatorial optimization problems. Our approaches are based on a combination of global–local heuristics. A fast binary Hopfield neural network (HNN) is used as local search approach to solve the WBP constraints. A genetic algorithm (GA) or a simulated annealing (SA) algorithm are hybridized with the HNN as global search approaches, to improve the quality of the solutions found by the HNN. We evaluate the performance of these approaches in several computational WBP instances, and in a communications network design application.

The structure of the rest of the paper is the following: in Section 2, we provide the WBP model we use, and based on it, we present the definition of the problem. It includes a subsection in which we provide an analysis of the WBP, showing its possible application to the design of mobile telecommunication networks. We present the algorithms for solving the WBP in Section 3, they are two hybrid approaches, which include a Hopfield network as local search algorithm for solving the problem's constraints, and a genetic algorithm or a simulated annealing as global search heuristics. In Section 4, we show the performance of these approaches. Section 5 concludes the paper giving some final remarks.

2. The WBP: model and definition

We consider the following model for the WBP: Let G be the set of guests to the banquet, formed by L guests, and T the set of M tables available. Each table i, $i = 1, \ldots, M$ has a maximum capacity of η_i people. Note that you cannot assign more than η_i guests to table i. Let H be an $L \times L$ matrix of *relationship* between guests, such that the element h_{ij} is a measure of how is the relationship between guests i and j. If these two guests have a good relationship h_{ij} will have a high value, but if the guests do not get on well, the value of h_{ij} will be low. In addition, let A be an $M \times L$ binary matrix of assignment, where every element $a_{ij} = 1$ stands for the guest j has been assigned to table i, whereas $a_{ij} = 0$ means no assignment. Let us consider now a partition of the set G of guests into N groups of people R_k, $k = 1, \ldots, N$, and let P_{R_k} be the number of guests in group R_k. Let $\hat{A}$ be an $M \times N$ binary matrix of assignment, where

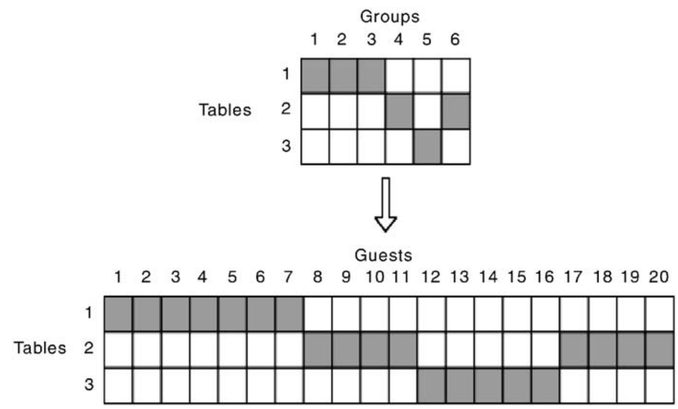

Fig. 1. Expansion step from a matrix $N \times M$ to a matrix $L \times M$ previous to the calculation of the objective function.

the element $\hat{a}_{ij} = 1$ means that group j has been assigned to table i. It is easy to see that the following constraint must be satisfied:

$$\sum_{k=1}^{N} P_{R_k} = L, \quad (1)$$

note also that if we know the amount of people in each group (P_{R_k}), we can generate matrix A from matrix $\hat{A}$. See Fig. 1 as an example of this expansion step.

The WBP can be mathematically defined as find an assignment matrix $\hat{A}$, such that the associate matrix A fulfills

$$\max\left(\sum_{i=1}^{M}\sum_{j=1}^{L}\sum_{k=1}^{L} a_{ij}a_{ik}h_{jk}\right) \quad (2)$$

subject to:

$$\sum_{j=1}^{N} \hat{a}_{ij} P_{R_j} \leqslant \eta_i \quad \forall i. \quad (3)$$

In the general case, the WBP considers that always is possible the assignment of groups to tables, without splitting the groups. Note that there are situations in which this is not possible, and the only possibility for assigning the groups to tables is splitting them into smaller groups. In practice, this situation is not very common, so in this paper we focus in the general case.

2.1. WBP in mobile network design

One of the most important considerations regarding the WBP, is that it can be seen as a generalization of the TA, (Abuali et al., 1994; Khuri and Chiu, 1997), where some of the terminals are assigned together. This implies a reduction of the problem encoding, from a matrix of encoding A to a matrix of encoding $\hat{A}$. Thus, the search operators of any metaheuristic approach for solving the WBP will be applied in the search space of matrices $\hat{A}$, much smaller than the size of the space of matrices A. This

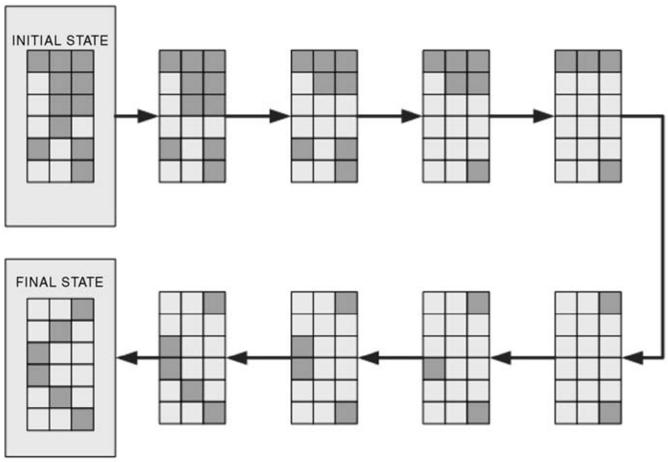

Fig. 2. Example of the Hopfield neural network implementation for the example in Section 3.1.1.

approach is applicable in several fields of engineering. For example, there is a direct application of the WBP in the design of the access part of a mobile communications network (Fig. 2).

Consider a mobile communications network formed by a set of BTSs, which provide mobile telecommunications services over an specific given area. Some of these BTSs can serve as *multiplexers*, concentrating the traffic handled by a group of BTSs, and forming a second level in the physical layout of the network. The highest level in the access part of a mobile network is formed by the so-called base stations controllers (BSCs), which perform the radio resource management, and take part in the mobility management and call control of the network. It can be considered as the first smart element in the network (Pomerleau et al., 2003; ETSI, 1996). Fig. 3 shows an example of a mobile communications network containing BTSs, multiplexers and BSCs.

The physical network architecture we have explained above, using multiplexers equipment, are specially useful in rural areas, where users in the BTSs generate low traffic levels, so it is not worth to establish direct links between the BTSs and the BSCs. Furthermore, in these rural areas, the distance between BTSs and BSCs are relatively large, which affects the reliability of the radio links. Thus, the problem we face consists of, given a group of BTSs[1] (grouped by the nearest multiplexer), and a set of BSCs, assigning the groups of BTSs to BSCs in such a way that a constraint of capacity to be fulfilled, and a given objective function to be maximized or minimized. It is easy to see that this corresponds to a WBP, where the guests are the BTSs, the groups of guests are groups of BTSs associated to a multiplexer, and the tables are the BSCs. The definition of a given BTS as a multiplexer can be done using the matrix H of relationship between guests, defined as k/d_{ij}, where k is a positive constant, and d_{ij} stands for the Euclidean distance between BTSs i and j. This matrix could be used as well for defining an objective function for the problem.

[1]Note that a group can be formed by one or more BTSs.

3. Proposed approach to the WBP: two hybrid metaheuristics

In this Section, we describe the meta-heuristic approaches for the WBP that we propose in this paper. Basically, we compare the performance of two hybrid metaheuristics, a Hopfield network-genetic algorithm approach (WBP-HNN_GA), and a Hopfield network-simulated annealing algorithm (WBP-HNN_SA). We describe the hybrid approaches, presenting the HNN used and the global search heuristics.

3.1. The Hopfield neural network

The Hopfield network we use as a local search algorithm for solving the WBP constraints is a binary Hopfield network (Shrivastava et al., 1992), where the neurons can only take values 0 or 1. The dynamics of this network depends on a matrix C which defines the minimum distance between two is in the network for each row, and on the initial state of the neurons. See (Shrivastava et al., 1992; Salcedo-Sanz and Bousoño-Calzón, 2005) for further details. The structure of the HNN can be described as a graph, where the set of vertices are the neurons, and the set of edges define the connections between the neurons. We map a neuron to every element in the solution matrix $\hat{A}$. In order to simplify the notation, we shall also use matrix $\hat{A}$ to denote the neurons in the network. The HNN dynamics can then be described in the following way: After a random initialization of every neuron with binary values, the HNN operates in a serial mode. This means that only one neuron is updated at a time, while the rest remain unchanged. Denoting by $\hat{a}_{ij}(t)$ the state of a neuron at time t, and letting $\pi(i)$ be a random permutation of $i = 1, 2, \ldots, N$, the updating rule of the HNN is defined as follows:

$$\hat{a}_{\pi(i)j}(t) = \mathrm{isgn}\left(\sum_{\substack{p=1 \\ p \neq \pi(i)}}^{N} \sum_{\substack{q=\max(1, j-c_{\pi(i),p}+1) \\ q \neq j}}^{\min(M, j+c_{\pi(i),p})} \hat{a}_{pq} \right), \quad (4)$$

where the isgn operator is defined by

$$\mathrm{isgn}(n) = \begin{cases} 0 & \text{if } n > 0 \text{ or } \sum_{j=1}^{N} \hat{a}_{ij} P_{R_j} \geqslant \eta_i \quad \forall i, \\ 1 & \text{otherwise.} \end{cases} \quad (5)$$

This updating rule only takes into account neurons $\hat{a}_{pq}$ with value 1 within a distance of c_{ip}. The matrix C is an $N \times N$ matrix which encodes the problem's constraints, and it is defined as follows:

$$c_{ij} = \begin{cases} M & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Each element c_{ij} stands for the minimum separation between 1s in matrix $\hat{A}$. Thus, the matrix C we have

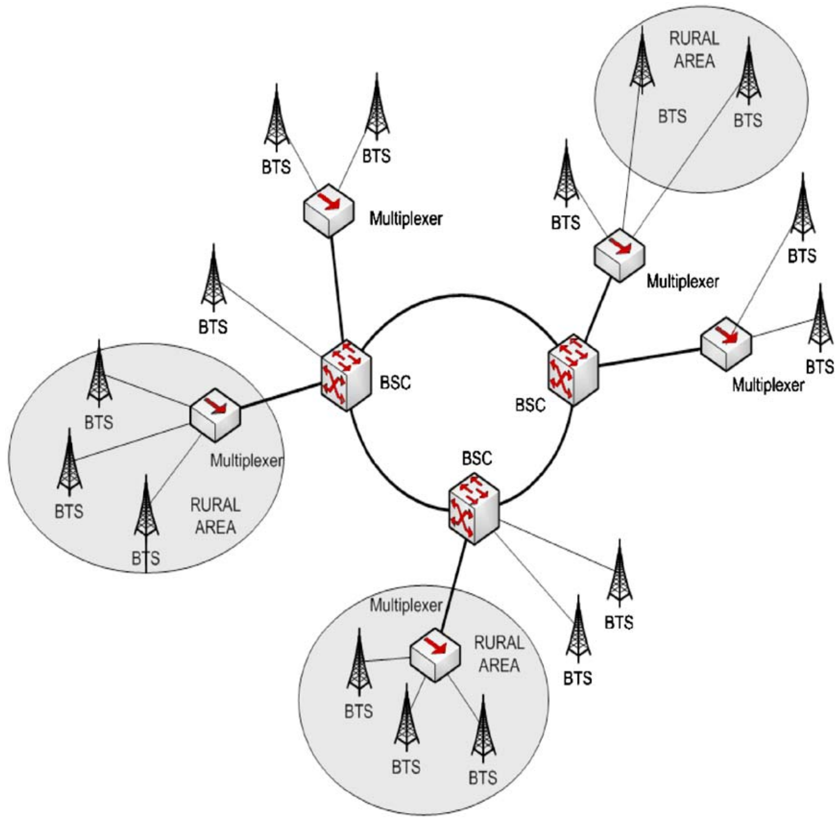

Fig. 3. Example of a mobile communications network containing BTSs, multiplexers and BSCs.

defined in Eq. (6) forces one and only one 1 per row (the minimum possible separation between two 1s in the same row is M), but there may be several 1s in the same column (0 separation if the 1s are in different rows of $\hat{A}$).

A *cycle* is defined as the set of $N \times M$ successive neuron updates in a given order. In a cycle, every neuron is updated once following the given order $\pi(i)$, which is fixed during the execution of the algorithm. After every cycle, the convergence of the HNN is checked. The HNN is considered converged if none of the neurons have changed their state during the cycle. The final state of the HNN dynamics is a potential solution for the WBP, which fulfils the problem's constraints given in Section 2. Note, however, that the solution found may be unfeasible if any of the groups is not assigned to a table.

100	100	88	19	78	64	1	91	13	31	3	26	46	51	1	38	67	97	77	16
100	100	51	78	77	69	83	55	9	46	55	73	21	4	32	80	99	37	32	88
88	51	100	100	100	56	20	38	56	35	7	85	44	16	28	26	23	48	6	33
19	78	100	100	100	84	79	72	36	39	29	69	51	44	62	39	99	54	5	81
78	77	100	100	100	77	28	82	74	73	95	85	51	85	4	80	19	46	66	56
64	69	56	84	77	100	100	8	10	10	5	56	96	61	57	96	49	68	58	64
1	83	20	79	28	100	100	19	83	1	93	12	17	6	95	15	84	69	34	46
91	55	38	72	82	8	19	100	100	100	100	54	54	49	40	15	66	54	13	90
13	9	56	36	74	10	83	100	100	100	100	76	76	17	84	31	40	34	18	17
31	46	35	39	73	10	1	100	100	100	100	83	13	31	20	93	1	91	47	54
3	55	7	29	95	5	93	100	100	100	100	76	14	46	6	37	16	92	83	37
26	73	85	69	85	56	12	54	76	83	76	100	100	100	100	100	10	82	53	37
46	21	44	51	51	96	17	54	76	13	14	100	100	100	100	100	61	8	18	69
51	4	16	44	85	61	6	49	17	31	46	100	100	100	100	100	24	28	93	57
1	32	28	62	4	57	95	40	84	20	6	100	100	100	100	100	69	37	25	17
38	80	26	39	80	96	15	15	31	93	37	100	100	100	100	100	13	38	81	70
67	99	23	99	19	49	84	66	40	1	16	10	61	24	69	13	100	100	100	100
97	37	48	54	46	68	69	54	34	91	92	82	8	28	37	38	100	100	100	100
77	32	6	5	66	58	34	13	18	47	83	53	18	93	25	81	100	100	100	100
16	88	33	81	56	64	46	90	17	54	37	37	69	57	17	70	100	100	100	100

Fig. 4. Example of matrix H for the Problem #1.

3.1.1. Implementation example

In this section we provide an example of how the binary Hopfield neural network used in this paper works. Consider a small WBP example formed by 20 guests, grouped in 6 groups, and 3 tables. The groups of guests for this example are defined to be $PR_i = \{2, 3, 2, 4, 5, 4\}$ (note that the sum of the groups is 20, the total number of guests), whereas the matrix H of happiness for this problem is showed in Fig. 4. Following Eq. (6), the constraints matrix C associated to this problems is

$$C = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}.$$

As was stated before, this matrix C does not allow more than one 1 in each row (each group must be assigned only to one table), but there may be any number of 1s in each column (several groups may be assigned to the same table) .

We are interested now in showing the process of reaching a feasible solution starting from an unfeasible one. Fig. 2 shows this process. The figure shows each state of the HNN (matrix $\hat{A}$), the shadowed squares in represent 1s in the HNN. We suppose that the initial unfeasible solution has been generated at random. In order to apply Eq. (4) to the initial state in the figure, we need to select an updating ordering in rows and columns. We choose π, which defines the order of updating, to be $\pi = \{4, 3, 5, 2, 1, 6\}$. For coming up with each HNN state, we apply Eq. (4) and the isgn operator to the previous state, until a feasible solution is obtained.

It is easy to check that in this case it takes only two cycles to converge to the final solution from the initial one. Note that, once the initial state of the network and the order of updating (π) are fixed, the final state is defined. Different initial states or updating orderings will produce different final solutions. It is also important to note that this Hopfield neural network does not take the objective function into account, but only the problem's constraints, defined by matrix C and isgn operator, in this case.

3.2. The hybrid approach WBP-HNN_GA

The first hybrid metaheuristic we consider for the WBP is formed by the Hopfield neural network described above, and a standard GA for improving the quality of its solutions. Our GA encodes a population of Υ potential solutions for the WBP, as binary strings of length $N \times M$. Each string represents a different assignment matrix $\hat{A}$ which is passed through the Hopfield network in order to get a feasible assignment of groups to tables. The population is evolved through successive generations by means of the application of the standard genetic operators selection, crossover and mutation described in Goldberg (1989). In addition, we implement an elitist strategy, which consists of passing the individual with the highest fitness to the next generation. This way, our algorithm always preserve the best solution found in the evolution. The complete algorithm for the WBP, formed by the GA and the HNN described in Section 3.1, is summarized below:

WBP-HNN_GA *algorithm*
Initialize GA population at random
while(max. number of generations has not been reached) **do**
  for(every individual $\hat{A}$)
    Run the HNN to obtain a feasible $\hat{A}$.
    Calculate the fitness value of the individual $f(\hat{A})$ using (2).
    if $\hat{A}$ is not feasible, apply a penalty to $f(\hat{A})$.
    Substitute the GA individual by the new $\hat{A}$ obtained through the HNN.
  endfor
  selection
  crossover
  mutation
end(while)

3.3. The hybrid approach WBP-HNN_SA

SA has been widely applied to solve combinatorial optimization problems (Kirpatrick et al., 1983; Kirpatrick, 1984; González et al., 2002). It is inspired by the physical process of heating a substance and then cooling it very slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches to an equilibrium point, and no more changes occur. Each stage of the process consists of changing the configuration a given number of times, until a thermal equilibrium is reached, and then a new stage starts, with a lower temperature. The solution of the problem is the configuration obtained in the last stage. In the standard SA, the changes in the configuration are performed in the following way: A new configuration is built by a random mutation of the current one. If the new configuration is better, then it replaces the current one, and if not, it may replace the current one probabilistically. This probability of replacement is high in the first stages of the algorithm, and decreases in every stage.

In this paper we consider the hybridization of a SA and the Hopfield neural network presented in Section 3.1 for solving the WBP. The idea behind this is that configurations involved in the SA are feasible solutions for the WBP. The SA will then search for the best feasible solution with respect to a given cost function, in this case a non-standard cost function for the WBP given in (2).

The complete algorithm for the WBP, formed by the SA and the HNN described in Section 3.1, performs in the following way:

WBP-HNN_SA *algorithm:*
$k = 0$;
$T = T_0$;
Initialize a potential solution at random;
Run the HNN to obtain $\hat{A}$;
evaluate($\hat{A}$,$f(\hat{A})$);
if $\hat{A}$ is not feasible, apply a penalty to $f(\hat{A})$.
repeat
for $j = 0$ **to** ξ
$\hat{A}_{\text{mut}} =$ **mutate**($\hat{A}$);
Run the HNN to obtain $\hat{A}$;
evaluate($\hat{A}_{\text{mut}}$,$f(\hat{A})$);
if $\hat{A}$ is not feasible, apply a penalty to $f(\hat{A})$.
if(($f(\hat{A}_{\text{mut}}) < f(\hat{A})$) OR ($random(0,1) < e^{(-\alpha/T)}$)) then
$\hat{A} = \hat{A}_{\text{mut}}$;
endif
endfor
$T = f_T(T_0, k)$;
$k = k + 1$;
until($T < T_{\min}$);

where k stands for the number of iterations performed; T keeps the current temperature; T_0 is the initial temperature; $T_{\min}$ is the minimum temperature to be reached; $\hat{A}$ stands for the current configuration and $\hat{A}_{\text{mut}}$ for the new configuration after the mutation operator is applied; f represents the cost function considered (see formula (2)); ξ is the number of mutations performed for a given temperature T; f_T is the freezer function; and α is a constant. Parameter α and the initial temperature T_0 are chosen to have an initial acceptance probability about 0.8, a value usually used. The freezer function is defined as

$$f_T = \frac{T_0}{1+k}. \tag{7}$$

Finally, the minimum temperature $T_{\min}$ is calculated on the basis of the desired number of iterations (numIt) as

$$T_{\min} = f_T(T_0, \text{numIt}). \tag{8}$$

4. Computational experiments

4.1. Synthetic WBP test instances

In order to test the performance of our approaches, we tackle a set of synthetic WBP instances of different sizes. Table 1 shows the main characteristics of the problems solved. There are six problems, of increasing difficulty. The matrices of relationship between guests, H, were randomly generated, assigning a value of 100 to the component h_{ij} if the guests i and j belong to the same group, and a random value between 0 and 99, extracted from a uniform probability distribution, to the rest of the components h_{ij}. The groups were also randomly generated, in such a way that the constraint 1 is fulfilled. Fig. 4 shows an example of an H matrix, for test Problem #1.

Table 1
Main characteristics of the instances tackled

Problem #	Guests	Groups	Tables	Tables' capacity
1	20	6	3	8
2	40	15	5	10
3	60	20	8	8
4	80	25	10	9
5	100	30	15	7
6	150	45	20	8

We have run the two meta-heuristic algorithms proposed using the following parameters: the GA was run with a population of $\Upsilon = 50$ individuals, evolving during 300 generations. We use two-point crossover, with a probability $P_c = 0.6$, a standard random flip mutation, with probability of mutate and individual equal to $P_m = 0.01$ and roulette wheel selection, see Goldberg (1989) for details on standard GA parameters. The main parameters of the SA are $\xi = 50$ and numIt = 300, with standard random flip mutation for obtaining new configurations. Note that the number of function evaluations is the same for our two metaheuristics approaches (15 000). In order to compare the results obtained by our approaches with an existing algorithm, we have implemented the GA described in Khuri and Chiu (1997) for the TA. It is a GA with integer encoding and penalty function to manage the unfeasible solutions (see Khuri and Chiu (1997) for details). The parameters of this GA are the same parameters than the GA of our hybrid heuristic. We run each algorithm 30 times, keeping the best, mean and standard deviation results obtained.

4.2. Results on synthetic instances

Table 2 shows the results obtained on the problems considered. It shows the best, average and standard deviation values of the 30 experiments run in every problem. Our proposed hybrid approaches are able to solve the problem, obtaining good quality solutions. The data in this table shows that the hybrid approach WBP-HNN_SA outperforms the hybrid WBP-HNN_GA in the majority of the instances. It is easy to see that both heuristics outperform the GA with penalty function. Table 3 shows the results of a t-test performed over the data obtained by the three compared algorithms. This table shows that the differences been our hybrid algorithms are statistically significant in Problems #4–6, whereas in instances #2 and #3 none of the approaches perform statistically better than the other. Both algorithms obtain the optimal assignment in the easiest instance #1. This table also shows that both hybrid heuristics perform statistically better than the GA with penalty function, in all instances but in the easiest one, instance #1, where all

Table 2
Comparison of the results obtained by the different algorithms considered

Problem #	GA (Best/Avg./Std Dev.)	WBP-HNN_GA (Best/Avg./Std Dev.)	WBP-HNN_SA (Best/Avg./Std Dev.)
1	9342/9342.0/0.0	9342/9342.0/0.0	9342/9342.0/0.0
2	23 432/22 328.3/680.1	24 056/23 207.1/349.7	23 838/23 205.1/183.2
3	28 345/27 302.3/725.8	30 118/29 562.0/272.4	29 760/29 466.4/328.7
4	44 322/43 600.2/816.4	45 314/44 268.4/466.1	45 488/44 973/613.3
5	44 913/44 298.2/746.3	45 902/45 080.2/518.6	46 592/46 178.4/222.9
6	73 100/72 493.8/694.1	76 920/75 454.1/914.6	77 030/75 886.7/628.5

Table 3
t values obtained by a two-tailed t-test for Problems #1 to #6

Problem #	GA-(WBP-HNN_GA)	GA-(WBP-HNN_SA)	(WBP-HNN_GA)-(WBP-HNN_SA)
1	0.0	0.0	0.0
2	−4.32[a]	−4.28[a]	0.02
3	−7.40[a]	−6.90[a]	1.58
4	−9.17[a]	−11.31[a]	−4.65[a]
5	−4.51[a]	−12.40[a]	−9.84[a]
6	−7.82[a]	−8.66[a]	−2.45[a]

[a]stands for values of t with 29 degrees of freedom which are significant at $\alpha = 0.05$.

the heuristics tested obtained the optimal solution in all the runs.

Table 4 shows the performance of our hybrid metaheuristics in the test problems #1–6 when no grouping is considered in the definition of the WBP (or, equivalently, L groups of size 1 are considered). In this case, both metahyheuristics encodings are larger than in the previous experiments. Note that the performances of WBP-HNN_GA and WBP-HNN_SA algorithms with this encoding are worse than using a encoding based on groups of guests. It is specially dramatic in the hardest Problems #4–6, where the differences in performance considering or not groups of guests are huge. This fact can be explained considering Fig. 5. It shows the differences in size of the metaheuristics encodings with and without groups of guests. As an example, Problem #4 can be encoded with 250 bits using groups ($N \times M$), whereas 800 bits ($L \times M$) are needed if no group of guests is considered. It is easy to see that these differences in size encoding affects to the metaheuristics convergence and performance.

In order to understand the role of the global search heuristics (GA and SA) in the algorithm, we can compare the results obtained by the HNN working with and without global heuristics. To do this, we launched 15 000 HNNs for each problem (without global search heuristics), keeping the best and average values obtained. Table 5 shows these results. The solutions obtained by the HNN are poor quality ones compared with the solutions obtained by the hybrid approaches WBP-HNN_GA and WBP-HNN_SA (see Table 2). Note that the differences between the HNN without global search heuristics and the hybrid approaches

Table 4
Comparison of the results obtained by the hybrid algorithms considered when no grouping in the encoding of the metaheuristics is considered

Problem #	WBP-HNN_GA (Best/Avg.)	WBP-HNN_SA (Best/Avg.)
1	9342/8974.2	9342/8854.1
2	22 140/21 576.3	21 859/20 998.3
3	27 418/26 382.0	26 754/25 754.9
4	41 325/40 208.3	40 953/39 525.3
5	42 117/41 198.3	42 342/41 339.8
6	70 654/69 018.1	71 101/69 823.4

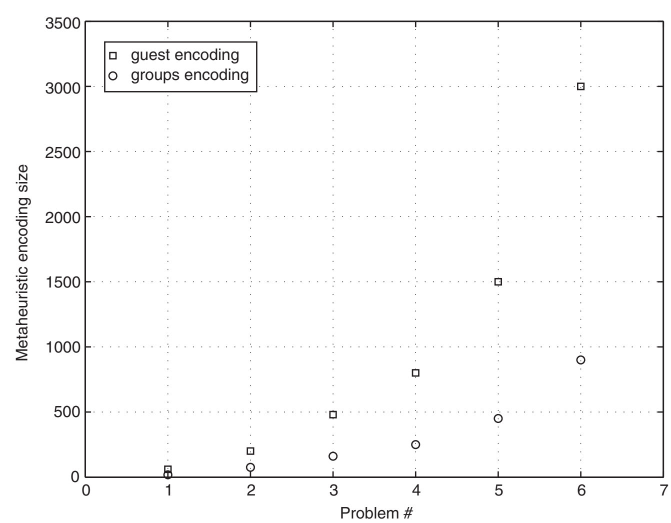

Fig. 5. Differences in size encoding using and not using groups of guests for the WBP test instances considered.

Table 5
Comparison of the results obtained by the HNN without hybridization with any global search heuristic

Problem #	HNN (Best/Avg.)
1	9342/8516
2	23 750/19 032.1
3	28 872/26 930.0
4	44 080/39 800.9
5	45 082/42 903.5
6	74 784/69 961.6

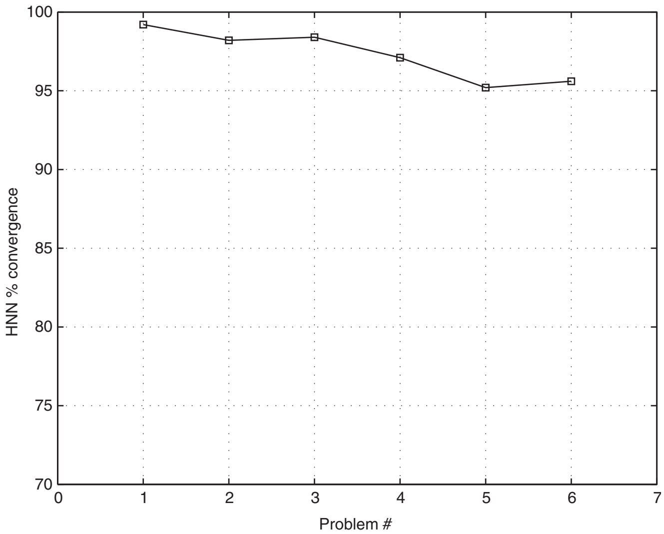

Fig. 6. HNN percentage of convergence to feasible solutions in the WBP test instances considered.

are more pronounced in the large size problems than in the small size ones. In average, the HNN without global search heuristic obtains poor solutions to the WBP. This is somehow expected, since the HNN does not take into account the objective function in its dynamics, but only the problem's constraints. On the other hand, the HNN is able to converge to feasible solutions fast. This can be seen in Fig. 6, which shows the percentage of convergence of the HNN. The percentage of convergence is calculated as the number of HNN runs which provide feasible solutions, divided by the total number of HNN launched (15 000 in this case). Note that its convergence is over 95% in all the instances considered. This point is important to hybridize the HNN with the GA or the SA algorithms guaranteeing the feasibility of the solutions.

4.3. An application of the WBP in mobile communications network design

In this Subsection, we show the application of the WBP to the design of a mobile communications network. Given the set of BTSs and controllers of the Fig. 7, and following the definitions in Section 2.1, the problem consists of assigning BTSs to controllers, in such a way that the sum of the distances between BTSs and the corresponding controllers to be minimum, and a constraint of capacity in the controllers is fulfilled. There are 80 BTSs and 5 controllers in our problem, which positions have been randomly generated in a 200×200 grid. We consider that once a BTS is assigned to a controller, the capacity of the controller is reduced one unit. Note that in this case, we consider the capacity of the BSCs as the number of network interfaces towards the BTSs, and not as the amount of traffic the BSC can handle. The capacity of each controller is fixed to be 17 for our simulation, so each controller can handle a maximum of 17 BTSs.

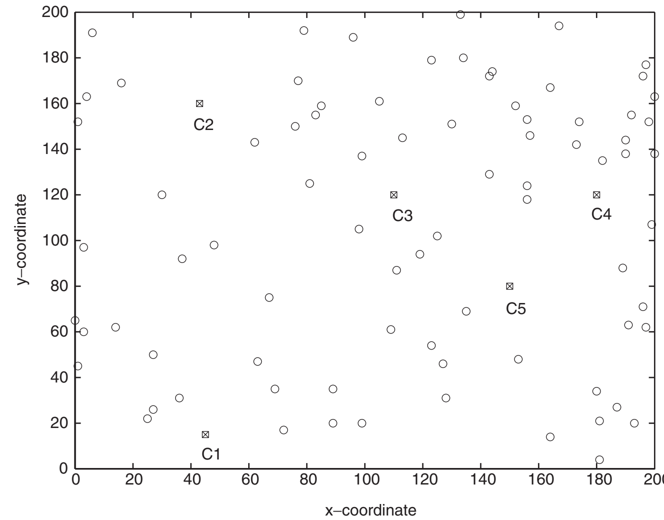

Fig. 7. Position of the BTSs (circles) and concentrators (squares) considered in the network design application of the WBP.

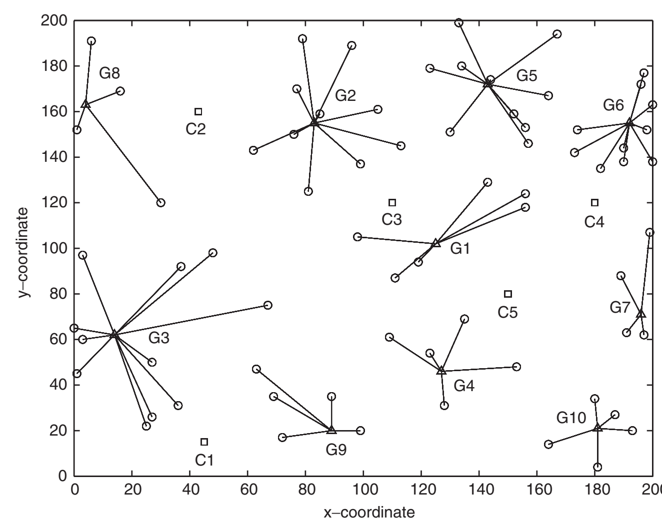

Fig. 8. Groups formed in the mobile network consider. Some BTSs serve as multiplexers (triangles).

Some of the BTSs in the network can serve as multiplexers (see Fig. 8), concentring the traffic generated by a group of BTSs and forming a second level in the management the network traffic. The number of multiplexers in a mobile communications network can vary a lot depending on the structure and size of the network. We fix the number of BTSs acting as multiplexers to $N = 10$ for our application.

It is easy to see the WBP structure in a mobile network design application like the one we consider above: the guests in the WBP are the BTSs in the network design problem, the groups in the WBP are associated to the multiplexers in the network and finally the tables of the WBP are the controllers in the network. Fig. 8 shows the position of the BTSs serving as multiplexers in our problem (triangles in the figure). In order to define these multiplexers' position, we define the matrix H following the indications given in Section 2.1 ($h_{ij} = k/d_{ij}$). The groups are formed then by obtaining N BTSs out of the total L, in such a way that the value of $\sum h_{ij},\ i = 1, \ldots, N,\ j = 1, \ldots, L$ is maximum. Note that all the guests in each group *get on well* with each other, i.e. the distance between them is the minimum possible. Note also that in this application, we use the matrix H of relationship between guests to form the groups.

Once the groups are formed, the WBP is completely defined by choosing an objective function. In this case, it is easy to see that the objective function to be minimized must be the distance between groups and tables (distance between multiplexers and controllers). As can be seen in Fig. 8, the WBP we must solve consists of 70 guests, 10 groups and 5 tables. We have solved the problem using both hybrid approaches presented in this paper, and both of them found the same solution to the problem, expressed as the matrix $\hat{A}^{\mathrm{T}}$ (T stands for transposition):

$$\hat{A}^{\mathrm{T}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

In this solution, groups 3 and 9 are assigned to Table 1, groups 2 and 8 are assigned to Table 2, groups 1 and 5 to Table 3, group 6 is assigned to Table 4 and finally groups 4, 7 and 10 are assigned to Table 5. It is easy to check that this assignment fulfils the capacity constraint of the tables, and it is optimal, in the sense that the sum of the distances from multiplexers to BSCs is the minimum possible.

5. Conclusions and future lines of work

In this paper we have presented an extension to the terminal assignment problem (TA), the so-called wedding banquet problem (WBP), which generalizes the TA in the case that groups of terminals must be assigned together. We have presented a mathematical definition for the WBP and introduced two hybrid metaheuristics algorithms for solving it. Both metaheuristics are based on a local–global search scheme, where the local algorithm is a Hopfield neural network which manages the problem's constraints. On the other hand, the global search algorithms try to improve the quality of the solutions found by the Hopfield network. We have tested as global search heuristics a genetic algorithm and a simulated annealing approach. The WBP have direct application to the design of mobile communication networks. We have described in the paper one of these applications in telecommunications. Other applications of the problem can be described and analyzed in the future.

Acknowledgements

The authors would like to thank the anonymous reviewers for their interesting comments and hints.

References

Abuali, F.N., Schoenefeld, D.A., Wainwright, R.L., 1994. Terminal assignment in a communications network using genetic algorithms. In: Proceedings of the 22nd Annual ACM Computer Science Conference. ACM Press, New York, pp. 74–81.

Brown, E., Sumichrast, R.T., 2005. Evaluating the performance advantages of grouping genetic algorithms. Engineering Applications of Artificial Intelligence 18, 1–12.

Brudaru, I., 2003. Grouping-based hybrid genetic algorithm for the terminal assignment problem. In: Proceedings of the Seventh International Research/Expert Conference Trends in the Development of Machinery and Associated Techniques. Lloret de Mar, Barcelona, Spain.

Chu, C.H., Premkumar, G., Chou, H., 2000. Digital data networks design using genetic algorithms. European Journal of Operational Research 127, 140–158.

ETSI, 1996. Digital Cellular telecommunications architecture (phase 2); network architecture (GSM 03.02). European Telecommunications Standards Institute.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

González, J., Rojas, I., Pomares, H., Salmerón, M., Merelo, J.J., 2002. Web newspaper layout optimization using simulated annealing. IEEE Transactions on Systems, Man and Cybernetics Part B 32 (5), 686–691.

Kershembaum, A., 1993. Telecommunications Network Design Algorithms. McGraw-Hill, New York.

Kirpatrick, S., 1984. Optimization by simulated annealing—quantitative studies. Journal of Statistical Physics 34, 975–986.

Kirpatrick, S., Gerlatt, G.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Khuri, S., Chiu, T., 1997. Heuristic algorithms for the terminal assignment problem. In: Proceedings of the 1997 ACM Symposium on Applied computing. ACM Press, New York, pp. 247–251.

Menon, S., Gupta, R., 2004. Assigning cells to switches in cellular networks by incorporating a pricing mechanism into simulated annealing. IEEE Transactions on Systems, Man and Cybernetics B 34 (1), 558–565.

Pomerleau, Y., Chamberland, S., Pesant, G., 2003. A constraint programming approach for the design problem of cellular wireless networks. In: Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, pp. 881–884.

Salcedo-Sanz, S., Bousoño-Calzón, 2005. A portable and scalable algorithm for a class of combinatorial optimization problems. Computers & Operations Research 32 (10), 2671–2687.

Salcedo-Sanz, S., Yao, X., 2004. A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. IEEE Transaction On Systems, Man and Cybernetics B 34 (6), 2343–2353.

Salcedo-Sanz, S., Bousoño-Calzón, C., Figueiras-Vidal, A.R., 2003. A mixed neural-genetic algorithm for the broadcast scheduling problem. IEEE Transactions, on Wireless Communication 2 (2), 277–283.

Shrivastava, Y., Dasgupta, S., Reddy, S.M., 1992. Guaranteed convergence in a class of Hopfield networks. IEEE Transactions on Neural Networks 3 (6), 951–961.

Smith, K., 1999. Neural networks for combinatorial optimisation: a review of more than a decade of research. INFORMS Journal on Computing 11 (1), 15–34.

Smith, K., Palaniswami, M., Krishnamoorthy, M., 1996. A hybrid neural approach to combinatorial optimisation. Computers & Operations Research 23 (6), 597–610.

Soni, S., Narasimhan, S., LeBlanc, L.J., 2004. Telecommunication access network design with reliability constraints. IEEE Transactions on Reliability 53 (4), 532–541.

Wang, G., Ansari, N., 1997. Optimal broadcast scheduling in packet radio networks using mean field annealing. IEEE Journal of Selective Areas Communication 15 (2), 250–259.

Yao, X., Wang, F., Padmanabhan, K., Salcedo-Sanz, S., 2004. Hybrid evolutionary approaches to terminal assignment in communications networks. in: Hart, W.E., Krasnogor, N., Smith, J. (Eds.), Recent Advances in Memetic Algorithms and related search technologies (invited book chapter). Springer, New York.