Comput Optim Appl (2015) 62:589–607
DOI 10.1007/s10589-015-9749-1

Tabu search and GRASP for the capacitated clustering problem

Anna Martínez-Gavara[1] · Vicente Campos[1] · Micael Gallego[2] · Manuel Laguna[3] · Rafael Martí[1]

Received: 30 July 2013 / Published online: 3 April 2015

Abstract The capacitated clustering problem (CCP) consists of forming a specified number of clusters or groups from a set of elements in such a way that the sum of the weights of the elements in each cluster is within some capacity limits, and the sum of the benefits between the pairs of elements in the same cluster is maximized. This problem—which has been recently tackled with a GRASP/VNS approach—arises in the context of facility planners at mail processing and distribution. We propose a tabu search and several GRASP variants to find high quality solutions to this NP-hard problem. These variants are based on several neighborhoods, including a new one, in which we implement a one-for-two swapping strategy. We also hybridize both methodologies to achieve improved outcomes. The maximally diverse grouping problem (MDGP) is a special case of the CCP in which all the elements have a weight of 1 U. This problem has been recently studied in the academic context when forming student groups, and we adapt the best method reported in the literature, a

✉ Rafael Martí
Rafael.Marti@uv.es

Anna Martínez-Gavara
gavara@uv.es

Vicente Campos
Vicente.Campos@uv.es

Micael Gallego
Micael.Gallego@urjc.es

Manuel Laguna
laguna@colorado.edu

1 Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Valencia, Spain

2 Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Móstoles, Spain

3 Leeds School of Business, University of Colorado at Boulder, Boulder, USA

tabu search with strategic oscillation (TS_SO), to the CCP. On the other hand, the handover minimization in mobility networks is a problem equivalent to the CCP in which we minimize the sum of the benefits (costs) of the edges between different clusters. GRASP with Path Relinking has been recently applied to it. Our empirical study with 133 instances shows the superiority of the new GRASP with tabu search for the CCP with respect to these three previous approaches: the GRASP/VNS, the adapted TS_SO, and the GRASP with Path Relinking.

Keywords Capacitated clustering · Diversity problems · Tabu search · GRASP · Graph partitioning

1 Introduction

Given a graph $G = (V, E)$ where V is a set of n nodes and E is a set of edges, let $w_i \geq 0$ be the weight of node $i \in V$ and let c_{ij} be the benefit of edge $(i, j) \in E$. The capacitated clustering problem (CCP) consists of partition V into p clusters in such a way that the sum of the weights of the elements in each cluster is within some integer capacity limits, L and U, and the sum of the benefits between the pairs of elements in the same cluster is maximized.

The CCP can be formulated as a quadratic integer program with binary variables x_{ik} that take the value of 1 if element i is in cluster k and 0 otherwise.

$$
\begin{array}{lll}
\text{(CCP) Maximize} & \displaystyle\sum_{k=1}^{p}\sum_{i=1}^{n-1}\sum_{j>i}^{n} c_{ij}x_{ik}x_{jk} & \\
\text{subject to} & \displaystyle\sum_{k=1}^{p} x_{ik} = 1 & i = 1, 2, \ldots, n \\
& L \leq \displaystyle\sum_{i=1}^{n} w_i x_{ik} \leq U & k = 1, 2, \ldots, p \\
& x_{ik} \in \{0, 1\} & i = 1, \ldots, n \; k = 1, \ldots, p
\end{array}
$$

The objective function adds the total benefit of all pairs of elements that belong to the same cluster. The first set of constraints forces the assignment of each element to a cluster. The second set of constraints forces the sum of the weights of the pairs of elements in the same cluster to be between L and U.

The literature on the CCP and related problems is vast and a recent paper [2] summarizes previous heuristics and formulations for this problem. We therefore do not duplicate the work here and refer the reader to Deng and Bard's excellent survey. We just mention that the CCP is also known as the *node capacitated graph partitioning problem*, and [6] were first to study it, proposing strong valid inequalities for a branch and cut algorithm. The objective in that version is to minimize the sum of the costs of the edges between different clusters, which is equivalent to the maximization of the costs (benefits) of the edges within clusters formulated above. These authors tested their algorithm on three applications: compiler design, finite element computations, and electronic design.

Deng and Bard [2] proposed a greedy randomized adaptive search procedure (GRASP) with variable neighborhood search (VNS) that, according to their com-

putational study, outperforms previous approaches. An interesting application of the problem arises in the context of facility planners at mail processing and distribution centers within the US Postal Service. In particular, the design of the zones to help rationalize the bulk movement of mail can be framed as a CCP. An interesting characteristic of their GRASP is that it constructs the customary *restricted candidate list* (RCL) using both nodes and edges, which to the best of our knowledge is an unconventional strategy [7]. This design triggered our interest on this problem, to compare this unconventional GRASP implementation with a standard one in which the RCL is formed with either nodes or edges but not both simultaneously. Deng and Bard [2] computational study does not consider general CCP instances and limits itself to the case in which the node weights, w_i, are equal to 1. This means that although they proposed a procedure for the CPP, their tests were performed on the special case of the maximally diverse grouping problem (MDGP). We extend their testing to include general CCP instances to compare several GRASP designs.

Morán-Mirabal et al. [13] proposed a GRASP with Path Relinking for the handover minimization in the context of mobility networks. As a mobile transceiver moves between areas, it may need to connect over time to several base stations. The transfer of connection from one base station to another is called a handover. Each base is connected to one radio network controller (RNC), which controls many of its operations, including its traffic and handover. Handovers between base stations connected to different RNCs tend to fail. The handover minimization problem is to assign base stations to RNCs such that RNC capacity is not violated, and the number of handovers between base stations connected to different RNCs is minimized. The set of base stations assigned to a RNC can be viewed as a cluster, and the minimization of handovers between different clusters is equivalent to the maximization of handovers within the same cluster. Therefore, this problem is equivalent to the CCP and we also compare our method with this previous heuristic.

When reviewing the recent clustering literature, we found an interesting connection not previously reported. In particular, the MDGP consists of grouping a set of elements into p mutually disjoint groups in such a way that the diversity among the elements in each group is maximized. The diversity among the elements in a group is calculated as the sum of the individual distance between each pair of elements. The objective of the problem is to maximize the overall diversity, i.e., the sum of the diversity of all groups, when the size of each group is within a specified range. Clearly, the MDGP is a special case of the CCP for which $w_i = 1$ for all node i, and the distance between each pair of nodes (i, j) is the benefit c_{ij}. Therefore, from the CCP formulation above, the MDGP can be formulated as:

$$\text{(MDGP) Maximize} \quad \sum_{k=1}^{p} \sum_{i=1}^{n-1} \sum_{j>i}^{n} c_{ij} x_{ik} x_{jk}$$

$$\text{subject to} \quad \sum_{k=1}^{p} x_{ik} = 1 \qquad i = 1, 2, \ldots, n$$

$$L \leq \sum_{i=1}^{n} x_{ik} \leq U \qquad k = 1, 2, \ldots, p$$

$$x_{ik} \in \{0, 1\} \qquad i = 1, \ldots, n \; k = 1, \ldots, p$$

The MDGP is called the k-partition problem in [5] and the equitable partition problem in [11]. It belongs to the family of diversity problems [3,8,12]. One of the most popular MDGP applications appears in the academic context of forming student groups [15]. Gallego et al. [9] proposed a tabu search with strategic oscillation, TS_SO, for the MDGP that outperforms the previous approaches for this problem. Recognizing the connection between MDGP and CCP, as part of our work, we also extend the TS_SO to tackle the general CCP.

The main contributions of our current development are:

- A new (simplified) GRASP for the CCP and a thorough comparison with Deng's and Bard's GRASP.
- A comparison with the GRASP with Path Relinking [13] originally proposed for the handover minimization.
- An adaptation of TS_SO to handle instances of the CCP.
- A new TS for the CCP to be hybridized with GRASP.
- Testing of all the methods on a new set of CCP instances.

The next section summarizes the Deng and Bard GRASP for the CCP and the Gallego et al. TS_SO for the MDGP. Section 3 describes the proposed new procedures. This is followed by the description of our extensive experimentation that includes 50 new CCP instances with $n = 82$, 240, and 480, and the 83 instances introduced by [13] for the handover minimization problem with $n = 20$, 30, 40, 100, 200, and 400. Statistical analysis shows the merit of our approach when compared to existing methods (including the adaptation of TS_SO to the CCP).

2 Existing CCP and adapted MDGP approaches

Deng and Bard [2] proposed a GRASP for the CCP, which as customary in GRASP implementations, alternates between two phases: construction and improvement. In the construction phase, the p clusters are first seeded with the heaviest weight edges algorithm (HWE), and then completed with a greedy randomized procedure. Specifically, the HWE identifies the p nodes with the largest weights and assigns them, separately, to the p clusters. The heaviest edges incident to these nodes are then identified, and their endpoints are assigned to the corresponding clusters. An alternative constructive method, labeled CMC, proposed by Deng and Bard was shown to be inferior to HWE and therefore we do not consider it here.

HWE produces clusters containing two nodes. At this point, a candidate list CL of elements is built to continue the construction process according to the GRASP methodology. In particular, CL is formed with the nodes and edges (pairs of nodes) that can be inserted into a solution cluster without exceeding the upper capacity limit U. For each node $i \in CL$, let $I(i,k)$ be the increase in the objective function if node i is added to cluster k. In other words, the sum of the benefits c_{ij} for all the nodes j already in cluster k. Similarly, let $I(e,k)$ be the increase in the objective function when edge e is added to cluster k (i.e., when its two endpoint nodes are in the cluster). The restricted candidate list RCL is formed with the best elements in CL according to this evaluation. This set of top candidates is computed as a fraction α of the number of elements in CL, where $\alpha \in]0, 1]$. Deng and Bard implemented a so-called reactive

GRASP in which effective values for this search parameter are self-adjusted as the iterations progress.

An important characteristic of this method is that the CL consists of both, nodes and edges. That is, the next element to be included in the solution may be either a single node or a pair of nodes (if an edge is selected). This is somewhat unconventional in GRASP, if one considers the implementations in the annotated bibliography by [7]. We will compare this design with a more conventional design in which the CL consists of either nodes or edges, but not both.

In the second phase of their GRASP, Deng and Bard proposed three different neighborhoods to improve a constructed solution x: $N_1(x)$, $N_2(x)$, and $N_3(x)$. Let V_k be the set of nodes in cluster k of this solution, and let W_k be the sum of the weights of the nodes in V_k (i.e., $W_k = \sum_{i \in V_k} w_i$); then, W_k must be within the capacity limits: $L \leq W_k \leq U$ for $k = 1, 2, \ldots, p$.

$N_1(x)$ is the result of extended insertion moves, in which a node i is moved from a cluster k to a cluster s. The node i is only removed from V_k if this cluster remains feasible after the move (i.e., if $L \leq W_k - w_i$), but if there is not enough capacity in cluster s for node i (i.e., if $W_s + w_i > U$), instead of discarding the move, a node j in V_s is moved from cluster s to a different one, say cluster t, with enough capacity for node j. The move is performed if the final solution is feasible, in other words, if $L \leq W_k - w_i$, $L \leq W_s + w_i - w_j \leq U$, and $W_t + w_j \leq U$.

$N_2(x)$ consists of edge insertions. Given an edge $(i, j) \in E$, if both nodes are in the same cluster k(i.e., $i, j \in V_k$), an edge insertion considers moving the nodes to another cluster s, as long as the resulting solution remains feasible. In mathematical terms, we move (i, j) from cluster k to cluster s if

$$L \leq W_k - w_i - w_j \quad \text{and} \quad W_s + w_i + w_j \leq U$$

This neighborhood also contains moves of edges with endpoints in two different clusters. In particular, given an edge $(i, j) \in E$ with $i \in V_k$ and $j \in V_s$, it examines moving i to cluster s, moving j to cluster k, or moving both nodes to another cluster t. In any case, only capacity-feasible moves are considered.

$N_3(x)$ implements a classical swap move, that is, one in which a node i is moved from a cluster k to a cluster s, and simultaneously a node j is moved from the cluster s to the cluster k. As in the other neighborhoods, the move is performed if the resulting solution is feasible. In mathematical terms:

$$L \leq W_k - w_i + w_j \leq U \quad \text{and} \quad L \leq W_s + w_i - w_j \leq U.$$

Deng and Bard proposed two improvement methods based on these three neighborhoods. The first one, called CNS, examines the neighborhoods in a sequential/cyclical fashion, terminating when no improvement is possible. The second one implements the so-called variable neighborhood descent, VND, in which the method switches from the current neighborhood N_u to the next one, N_{u+1}, when N_u cannot produce an improvement of the current solution. Moreover, if a better solution is obtained with N_u with $u > 1$, the method switches back to N_1. The VND terminates when N_3 is applied and no further improvement is possible. Finally, the authors implemented a

Randomized VND, called RNVD, in which the neighborhood to be searched in the next iteration is probabilistically selected, where the probability of selection is linked to the merit of each neighborhood as determined by the quality of the solutions found during the search.

In the computational testing, Deng and Bard [2] compared their designs and concluded that the combination of HWE with RVND resulted in the best overall performance. We use this variant for the purpose of comparison and call it Prev_GRASP. As a side note, Deng and Bard tested a Path Relinking post-processing strategy and concluded that its inclusion did not result in a significant improvement and therefore we have left it out of our experiments.

Gallego et al. [9] proposed a tabu search with strategic oscillation, TS_SO, for the MDGP. TS_SO is the state-of-the-art solution method for MDGPs. Empirical evidence shows that TS_SO outperforms the improvement method LCW [16] and the hybrid genetic algorithm LSGA [4]. As described in the previous section, the MDGP is a special case of the CCP. We briefly describe TS_SO, which is based on three steps: (1) construction of the initial solution, (2) neighborhood search and (3) strategic oscillation, and our adaptation of TS_SO to the CCP.

The construction step (GC, for greedy construction) for the MDGP starts by randomly selecting p nodes and assigning each of these elements to a separate group (cluster in our case). Therefore, at the end of the first step, each group has one node assigned to it. Then, the procedure performs $n - p$ iterations to assign the remaining unassigned elements to groups. In order to generate a feasible solution, the iterations are divided into two phases. In the first phase, the elements are assigned to groups that currently contain fewer elements than the desired minimum number of elements. In the second phase, the remaining elements are assigned to groups with a number of elements that is smaller than the desired maximum number of elements.

In our adaptation of GC to the CCP, the first phase adds a randomly selected node i to the cluster k that maximizes $I(i,k)$ and for which $W_k < L$. When all the clusters have a sum of weights of their nodes larger than or equal to L, the second phase proceeds in a similar way, by adding randomly selected elements to the cluster k that maximizes $I(i,k)$, without exceeding the upper capacity limit U.

Once a solution x has been constructed, the neighborhood search (phase 2 of the TS_SO method) is applied. The search neighborhood in TS_SO consists of all node insertions and swaps. Note that insertions and swaps are considered simultaneously instead of sequentially, as done by [2]. Moreover, the TS_SO implements simple insertions, involving only two groups, instead of the extended version (involving three groups) implemented in $N_1(x)$. We denote the simple insertion neighborhood as $N_0(x)$. In mathematical terms, at each step the method selects the best feasible move in $N_0(x) \cup N_3(x)$. TS_SO includes a short-term tabu memory [10] that enables the method to search beyond the first local optimal point. Specifically, when the local search reaches a point where no improving moves are available, the best non-improving move in $N_0(x) \cup N_3(x)$ is selected and executed (i.e., the best available move). At this point, elements that are moved from their current group to another are not allowed to move again for $tabuTenure = n/10$ iterations. The process terminates when $maxIter = n/2$ consecutive iterations have been performed without improving the best solution found during the search. TS_SO can be straightforwardly adapted to the

CCP by simply checking the feasibility of moves in terms of the sum of the weights of the nodes in each cluster instead of the cluster cardinality.

So far, we have assumed that feasibility is maintained through the search. However, the third phase of TS_SO—the *strategic oscillation* (SO) phase—explores solutions for which the group cardinality restrictions may be violated. In particular, the method applies the neighborhood search described above but the number of the elements in a group is allowed to be outside the specified limits by a certain amount so that ranges between 0 and so_{max}. This means that when $so > 0$ the search is allowed to visit infeasible solutions. To create the oscillation pattern, the value of so is reset to one after every successful application of the improvement method, otherwise so is increased by 1 U until it reaches so_{max}.

In our adaptation of the SO phase to the CCP, we consider that the sum of the weights of the elements in a cluster has to be within the following limits:

$$L - so \leq W_k \leq U + so$$

Note that this method does not guarantee that the final solution is feasible. When this happens, we apply a repair mechanism that consists of removing elements from clusters k for which $W_k > U$ and adding elements to clusters s for $W_s < L$. The elements are selected at random and the process continues until the weight constraints of the clusters are satisfied. We refer to the entire adaptation of TS_SO to the CCP as CCP_TS_SO.

Morán-Mirabal et al. [13] proposed a GRASP with Path Relinking for the handover minimization problem, which as shown above, is equivalent to the CCP. A randomized greedy algorithm constructs a solution one base station to RNC assignment at a time. RNCs are initially permuted at random and the algorithm scans the RNCs in the permuted order, dealing with only one RNC at a time. Let k be the current RNC being scanned. Base stations are assigned to RNC k until this RNC does not have enough leftover capacity to accept another base station. After each base station is assigned to an RNC, the RNC's available capacity is adjusted to reflect the assignment just made. Base stations are assigned to the RNC while the RNC has available capacity. After scanning all available RNCs, it may occur that not all base stations are assigned. In such a case, a repair procedure is applied to attempt to achieve feasibility.

Once the randomized greedy construction method produces an assignment vector, a local search algorithm attempts to improve the assignment by making changes on it. Specifically, Morán-Mirabal et al. [13] proposed three local search algorithms, move-1, move-max, and swap-2. The three algorithms scan the base stations in increasing order of their total traffic. For base station i, the procedure move-1 checks if there is any other RNC with enough capacity to accommodate i such that the reassignment from its current RNC to the other one reduces the total handover count. If such RNC is found, base station i is reassigned to it. In the case of move-1, the procedure is restarted at the first base station in the permutation (i.e. the base station with the smallest traffic), whereas in the case of move-max it proceeds to the next station in the permutation (i.e. the station with least traffic among those with more traffic than the just reassigned station). The procedure ends when the all base stations are scanned and no improving

move is found. In the case of swap-2, pairs of base station assignments are considered for swapping.

3 A simplified GRASP for the CCP

In our view, the GRASP implementation of [2] is overly complicated and a simplified version should be able to obtain solutions of similar quality. The best way to describe their method is that it is the result of hybridizing Reactive GRASP with VND with three different neighborhoods that are combined in a probabilistic fashion, where the neighborhoods consist of a set of compound moves that may change the node assignment of up to three clusters. In addition, the candidate list in the construction process includes both nodes and edges. In contrast, we propose a simplified GRASP implementation in which only nodes are candidates in the construction process and two simple neighborhoods are combined into a deterministic VND design.

Our GRASP starts by seeding the p clusters $V_1, V_2, \ldots, V_p$ with p randomly selected nodes. Then, we explore the clusters in lexicographical order assigning elements until all of them satisfy the lower bound constraint (i.e., until the sum of the weights of the nodes already assigned to the cluster is larger than or equal to L). To do so, the candidate set CL is formed with all the unassigned nodes and the value $I\,(i,k)$ is calculated for all pairs (i,k) of nodes and clusters. RCL_k—that is, the restricted candidate list of nodes for cluster k—is formed with all nodes i for which $I\,(i,k)$ is within a percentage $\alpha \in]0,1]$ of the maximum value I_{max} in CL:

$$RCL_k = \{i \in CL : I\,(i,k) \geq \alpha I_{max}\} \text{ where } I_{max} = \max_{i \in CL} I\,(i,k)$$

Our constructive method randomly selects an element in RCL_k, and performs the corresponding assignment. The method proceeds in this way, starting with cluster $k = 1$, and assigning elements to it until the lower bound constraint is satisfied ($L \leq W_1$ where $W_1 = \sum_{j \in V_1} w_j$). Then, the process moves to cluster 2 and proceeds in this way until the sum of the weights of the elements assigned to each cluster is larger than or equal to L.

In the following steps, the candidate set CL is formed with the pairs (i,k) with unassigned nodes i and those clusters k such that the sum of the weights of the elements already assigned to the cluster plus the weight of i is lower than or equal to U.

$$CL = \{(i,k) : 1 \leq k \leq p, W_k + w_i \leq U\} \text{ where } W_k = \sum\nolimits_{j \in V_k} w_j$$

The method proceeds to evaluate $I\,(i,k)$ for all (i,k) in CL build RCL with the (i,k) pairs with an evaluation within a percentage $\alpha \in]0,1]$ of the maximum value in CL, and select one pair at random. It stops when all the nodes have been assigned to clusters.

Once a solution x is obtained, we apply our improvement method, which consists of a deterministic VND based on two neighborhoods, $N_0\,(x)$ and $N_3\,(x)$. The method determines first a best neighbor x' of x in $N_0\,(x)$. If x' is better than x, then x is replaced with x' and the method searches now for the best neighbor in $N_0\left(x'\right)$, thus

performing a local search in N_0 while it improves the current solution. When the current solution x cannot be improved in N_0, then the method resorts to N_3 and determines the best neighbor x' of x in $N_3(x)$. If x' is better than x, then the method comes back to search in $N_0(x')$; otherwise the VND finishes. In short, the algorithm performs a local search for the best solution in N_0 and only resorts to searching N_3 when the process is trapped in a local optimum found in N_0. The improvement method considers only feasible moves.

4 Tabu search for the CCP

As an alternative to the methods described in the previous sections, we engaged in the development of a tabu search specifically designed for the CCP. In other words, we studied the characteristics of the CCP and instead of adapting an existing procedure, we designed an original one. The main characteristic of this procedure consists of a new neighborhood structure based on 2-1 exchanges that we refer to as $N_4(x)$. The method is applied from the initial solution described in the previous section.

$N_4(x)$ may be considered a variant of $N_2(x)$ in the sense that it explores exchanges of two nodes, say i and j, belonging to the same cluster k for a node l that belongs to a cluster s. In this neighborhood, unlike in $N_2(x)$, nodes i and j may or may not be connected. That is, $N_4(x)$ does not require that $(i, j) \in E$. The neighborhood includes moves that are contained neither in $N_0(x)$ nor in $N_3(x)$. For instance, consider the following situation.

Cluster A contains nodes 1 and 2 with weights $w_1 = w_2 = 3$ and has remaining capacity of 2 U. A Cluster B contains node 3 with $w_1 = 6$ and remaining capacity of 1. Under $N_0(x)$, the insertions of node 1 to cluster B, node 2 to cluster B and node 3 to cluster A are considered. However, none of these moves are feasible. Likewise, the swaps of nodes 1 with 3 and 2 with 3 under $N_3(x)$ result in infeasible solutions. However, the 2-1 exchange that moves nodes 1 and 2 from cluster A to B and node 3 from cluster B to A is feasible.

Our short-term memory tabu search based on $N_4(x)$ operates as follows. All 2-1 exchanges are evaluated and the best move (according to the objective function value) is selected. The three nodes participating in the exchange are made tabu-active. In subsequent iterations, a move is classified tabu if it contains one or more tabu-active nodes. Nodes remain tabu-active for $tenure$ iterations. The tabu status of a move is waved if the exchange leads to a solution that improves upon the incumbent.

5 Computational experiments

This section describes the computational experiments that we performed to test the effectiveness and efficiency of the procedures discussed above. The GRASP by [2], Prev_GRASP, and our adaptation to the CCP of the TS_SO originally proposed for the MDGP by [9], CCP_TS_SO, were implemented in Java SE 6. Our GRASP (Sect. 3), simply called GRASP, and the TS (Sect. 4) for the CCP were implemented in C. For random number generation in our methods, we use the rand() function in C and the random class in Java, both with a seed of 0. All experiments were conducted on an

Intel Core 2 Quad CPU Q 8300 with 6 GB of RAM, with the exception of the last one (corresponding to the results reported in Table 9), which was performed on an Intel Core 2 i7 @ 2Ghz.

We employed 50 instances in our experimentation. This benchmark set of instances, referred to as CCPLIB, is available at http://www.optsicom.es/ccp. The set is divided into two subsets:

1. *RanReal*—This set, originally proposed by [9] for the MDGP, consists of 40 $n \times n$ matrices in which the benefit values c_{ij} are real numbers generated using a Uniform distribution U (0, 1000). We have adapted these instances to the CCP by generating the node weights with a Uniform distribution U (0, 10). There are 20 instances with $n = 240$, $p = 12$, $L = 75$, and $U = 125$. The remaining 20 instances have $n = 480$, $p = 20$, $L = 100$, and $U = 150$.
2. *DB*—This set is based on the problem with $n = 82$ and $p = 8$ introduced by [2] in the context of mail delivery. We keep the benefit values as they appear in that original instances. However, instead of considering all the node weights equal to 1, as in Deng's and Bard's article, we randomly generate the node weights with a Uniform distribution U (0, 10) and thus changing the instances from MDGP to CCP. The set consists of 10 instances with $n = 82$, $p = 8$, $L = 25$, and $U = 75$.
3. *MM*—The benchmark set of 83 synthetic instances introduced in [13] for the handover minimization problem. It consists of five instances for each of the following combinations (b, r) of number of base stations and RNCs: (20, 5), (30, 5), (30, 10), (40, 5), (40, 10), (40, 15), (100, 15), (100, 25), (100, 50), (200, 15), (200, 25), (200, 50), (400, 15), (400, 25), (400, 50), and four instances with the following combinations: (20, 10), (30, 15). It is available at http://www.research.att.com/~mgcr/data/handover-minimization.

We have selected 15 representative instances with different characteristics to perform a preliminary experimentation to identify effective values for critical search parameters and to compare different designs. Specifically, we select 6 *RanReal* instances with $n = 240$, $p = 12$, $L = 75$, and $U = 125$, 6 *RanReal* instances with $n = 480$, $p = 20$, $L = 100$, and $U = 150$, and 3 *DB* instances with $n = 82$, $p = 8$, $L = 25$, and $U = 75$.

In our first preliminary experiment we test four different variants of the simplified GRASP method for the CCP described in Sect. 3. In particular, we consider four different values of the parameter α, namely, 0.2, 0.4, 0.6, and 0.8. We generated 100 different solutions to each instance with each GRASP variant. Table 1 shows, for each variant, the average objective function value, the average percent deviation from the best solutions obtained within this experiment, and the number of instances in which the method is able to match the best solutions obtained within this experiment (out of 15 instances). These measures are "local" in the sense that the best solutions are those found within the experiment. They are used because they allow us to discriminate among the procedures being tested and identify the better alternatives. However, as a point of reference, Table 1 also includes the average deviation achieved by each variant against the best-know solutions. This is a "global" measure in the sense that the best-known solutions are those found across all experiments in this paper, which as far as we know represents the best-known published solutions.

Table 1 GRASP with different values of α

Metric	α			
	0.2	0.4	0.6	0.8
Average objective function value	233397.5	233203.9	233392.9	233226.3
Number of best solutions	3	3	5	4
Local average deviation	0.36 %	0.44 %	0.36 %	0.43 %
Global average deviation	14.95 %	14.91 %	14.95 %	14.96 %
Score (worst = 45)	24	23	19	24

Additionally, Table 1 includes the Score statistic described in [14]. For each problem instance in an experiment, the Score of a method is the number of methods that found a solution that is strictly better than the one that the method being scored found. For a set of instances, the Score is the sum of the individual scores. Therefore, while the overall best score in any experiment is zero, the worst score depends on the number of alternative procedures and the size of the test set. Since there are 15 instances in the test set for this experiment and four different procedures (one for each value of alpha) then the worst possible score is 45 (i.e., 15 instances times 3 competing procedures).

Table 1 shows the slightly better performance (in terms of both number of best solutions and deviation) achieved by GRASP with the parameter α set to 0.6. This table also shows that, in general, the differences observed among these statistics related to the four variants tested are relatively small. This is particularly true in the deviations with respect to the overall best known solutions, which confirms that GRASP constructions on their own are not able to produce high quality solutions. The local measures, including the Score all favor the parameter setting with $\alpha = 0.6$.

An important question in the context of multi-start procedures in general, and GRASP in particular, relates to the origin of the high-quality solutions obtained after applying the improvement method. Specifically, we are interested in measuring the correlation between the objective function value of the constructed solutions and the value of the improved solutions (i.e., the local optima found after applying the improvement method). We would like determine whether high-quality local optima are related to high-quality constructions. If so, this could lead us to implementing filter mechanisms to save the computational time invested on attempting to improve somewhat poor constructions. To this end, we generated 100 solutions for each problem in our set of 15 representative instances (testing set) and computed the correlation coefficient between both the initial solution and the local optimum. The resulting correlation coefficient of 0.01 indicates that good local optima can come from initial solutions of any quality. Figure 1 shows this lack of correlation for 100 solutions of a representative instance in our test set. We observe that the points are scattered in the plane without any discernible pattern.

In our second preliminary experiment we compare GRASP (with α set to 0.6) with the Prev_GRASP on the test set of 15 instances. We run both methods for 60 s on each instance and report the results in Table 2.

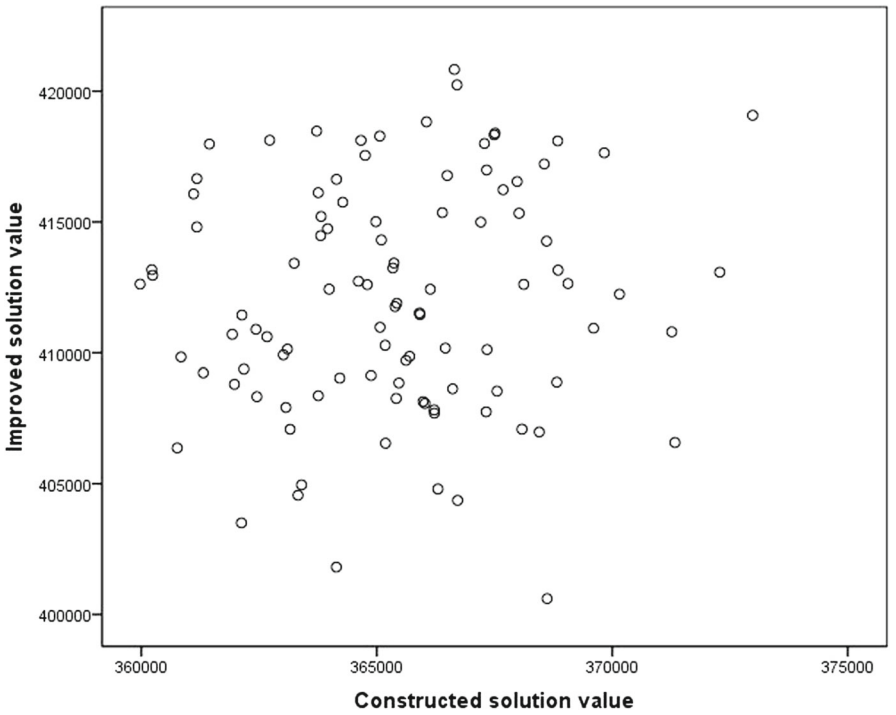

Fig. 1 Constructed and improved values of 100 solutions

Table 2 Comparison of GRASP methods

Metric	Prev_GRASP	GRASP(0.6)
Average objective function value	194,109.8	233,841.7
Number of best solutions	6	9
Local average deviation	10.99 %	2.84 %
Global average deviation	20.68 %	14.38 %
Score (worst = 15)	9	6

Table 2 shows that the proposed simplified GRASP seems to perform slightly better than the one in the literature. GRASP(0.6) is able to match 9 out of 15 best solutions and exhibits a 2.84 % deviation from the best solutions found within this experiment. The global average deviation (i.e., the one against the overall best-known solutions) of 14.38 % compares well with 20.68 % achieved by Prev_GRASP. Note that in a comparison of two methods, the Score provides the same information obtained by the "number of best solutions found." Although not shown in Table 2, it is interesting to point out that Prev_GRASP performs better than GRASP(0.6) on the medium size RanReal instances ($n = 240$), while GRASP(0.6) outperforms Prev_GRASP on the large RanReal instances ($n = 480$). We apply a statistical, non-parametric, test to confirm that both methods perform similarly. We employ the *Wilcoxon test* to make a

Table 3 Comparison of GRASP constructive variants

Metric	N_GRASP	E_GRASP	N&E_GRASP
Average objective function value	195495.8	189815.8	181338.4
Number of best solutions	15	0	0
Local average deviation	0.0 %	5.4 %	8.7 %
Global average deviation	28.1 %	32.3 %	34.6 %
Score (worst = 30)	0	16	29

Table 4 TS with different values of *tenure*

Metric	*tenure*			
	5	10	15	$\sqrt{n}$
Average objective function value	282305.3	282739.3	281972.7	281532.8
Number of best solutions	8	4	2	1
Local average deviation	0.29 %	0.88 %	2.05 %	1.03 %
Global average deviation	2.40 %	2.47 %	3.61 %	2.88 %
Score (worst = 45)	14	23	26	21

pairwise comparison of Prev_GRASP and GRASP(0.6). The results of this test (with a p value of 0.233) determined that the solutions obtained by the two methods could come from the same population.

To finish the analysis of GRASP methods, we compare three different designs for the construction phase: (1) the one used in Prev_GRASP in which the candidate list CL consists of both, nodes and edges (N&E_GRASP), (2) one in which the CL consists of nodes (N_GRASP), and (3) one in which the CL consists of edges (E_GRASP). To this end, we generated 100 solutions for each problem in our set of 15 representative instances with each of these three constructive methods and report, in Table 3, the results. They clearly show the superiority of the conventional GRASP construction, N_GRASP, in which the next element to be included in the solution is a single node, instead of an edge, or either a node or an edge.

In our fourth preliminary experiment, we test the tabu search method, TS, described in Sect. 4, with several $tenure$ values. We set this parameter to 5, 10, 15, and $\sqrt{n}$. Table 4 shows the results (i.e., average values of the five statistics) on the 15 test instances. These results show that the best outcomes are obtained with $tenure = 5$ since this variant is able to achieve eight best solutions and exhibits a local average percent deviation of 0.29 %.

In the final experiment of this block, we add to the comparison set our adaptation to the CCP of the tabu search developed by [9] for the MDGP. As described in Sect. 2, this method has the ability to search in the space of infeasible solutions by way of a strategic oscillation component. To test the merit of this feature, we have considered two variants of this adaptation: AdTS, without the SO, and AdTS_SO, with the oscillation. We also compare both methods with our TS with $tenure = 5$. In the variant with the

Table 5 Tabu search methods

Metric	AdTS	AdTS_SO	TS
Average objective function value	256,406.2	269,549.5	281,203.4
Number of best solutions	1	2	12
Local average deviation	6.81 %	2.89 %	1.06 %
Global average deviation	8.02 %	4.16 %	2.37 %
Score (worst = 30)	26	14	6

oscillation, AdTS_SO, following the recommendation in [9], we tested six values of the oscillation parameter: $so_{max} = 1, \ldots, 6$, and selected $so_{max} = 4$ since it obtains the best solutions. As in the previous preliminary experiments, we run each method for 60 s on each instance. The results associated with this experiment are in Table 5.

The results in Table 5 indicate that, as expected, our tabu search method specifically designed for the CCP obtains better solutions than the two versions adapted from the MDGP to the CCP. This table also shows that the strategic oscillation component (AdTS_SO) is effective when compared to the version without it. TS obtains 12 best solutions out of the 15 instances, while AdTS and AdTS-SO obtain 1 and 2, respectively. Additionally, both the local and global average deviations as well as the score favor TS.

We applied the Friedman test for paired samples to the data used to generate Table 5. The resulting p value of 0.001 obtained in this experiment indicates that there are statistically significant differences among the three methods tested (we are using the typical significance level of $\alpha = 0.05$ as the threshold between rejecting or not the null hypothesis). A typical post-test analysis consists of ranking the methods under consideration by their average rank values. The result is that the TS method is the best with an average rank of 2.60, followed by AdTS_SO (2.10), while AdTS ranks third (1.30). Finally, we compare the best two methods in this experiment, TS and AdTS_SO, with the pairwise Wilcoxon test. The resulting p value of 0.002 obtained confirms that the two methods are significantly different.

In the first experiment of the final block, we compare the best methods identified in the preliminary experimentation on the *RanReal* and *DB* sets totalizing 50 instances. Specifically, we compare:

- Prev_GRASP. The GRASP method by [2] described in Sect. 2.
- AdTS_SO. Our adaptation of the tabu search by [9] described in Sect. 2.
- GRASP. Our GRASP method described in Sect. 3.
- TS. Our tabu search method described in Sect. 4.
- GRASP + TS. A combination of the two methods that we have developed.

The GRASP + TS hybrid consists of running GRASP(0.6) for half of the total running time in the experiment, and then, for the rest of the running time, applying TS starting from the best solution found by GRASP.

Tables 6, 7, and 8, show respectively the results of the 20 *RanReal* instances with $n = 240$, 20 *RanReal* instances with $n = 480$, and the 10 *DB* instances. We run each method for 60 s on each instance.

Table 6 Comparison of best methods on the 20 *RanReal* instances with $n = 240$

Metric	Prev_GRASP	AdTS_SO	TS	GRASP	GRASP+TS
Global average deviation	9.43 %	2.79 %	0.15 %	14.55 %	0.61 %
Number of best solutions	0	0	16	0	4
Score (worst = 80)	60	40	4	80	16

Table 7 Comparison of best methods on the 20 *RanReal* instances with $n = 480$

Metric	Prev_GRASP	AdTS_SO	TS	GRASP	GRASP+TS
Global average deviation	41.78 %	7.54 %	1.99 %	19.85 %	2.07 %
Number of best solutions	0	0	10	0	10
Score (worst = 80)	80	37	10	60	13

Table 8 Comparison of best methods on the 10 *DB* instances

Metric	Prev_GRASP	AdTS_SO	TS	GRASP	GRASP+TS
Global average deviation	0.33 %	0.02 %	4.52 %	0.08 %	0.13 %
Number of best solutions	0	7	0	2	2
Score (worst = 40)	31	8	40	12	14

The results in Tables 6, 7, and 8 indicate that the performance of the competing methods varies with the data sets. For instance, Prev_GRASP has difficulties finding solutions of reasonable quality for *RanReal* problems with $n = 480$, producing a large average deviation of 41.78 %. However, the same method performs much better on the DB set, where it obtains an average deviation of 0.33 %. On the other hand, TS shows better performance than the other procedures in the *RanReal* sets but comes out last in the DB set.

We apply statistical tests to the data obtained from this last experiment in order to detect differences among the Prev_GRASP, AdTS_SO and GRASP+TS. From our three designs, GRASP, TS and GRASP+TS, we chose GRASP+TS because of its better average overall performance. We first applied the Friedman test to test the hypothesis that there are differences among the procedures. The hypothesis that they are the same is rejected with a p value lower than 0.001. The resulting ranking is GRASP+TS (2.74), AdTS_SO (2.19), and Prev_GRASP (1.07). We then applied the pairwise Wilcoxon test to GRASP+TS and AdTS_SO, with the outcome indicating a better performance of GRASP+TS. This conclusion is reached with a p value lower than 0.001.

In an attempt to understand how GRASP and TS interact within GRASP+TS, we perform a second experiment in this block. Specifically, we recorded the objective function value of the best solution found during the search when applying GRASP+TS to the first *RanReal* instance with $n = 240$. The profile of this run is depicted in Fig. 2. The plot shows that TS is able to provide a boost after the GRASP search stalls.

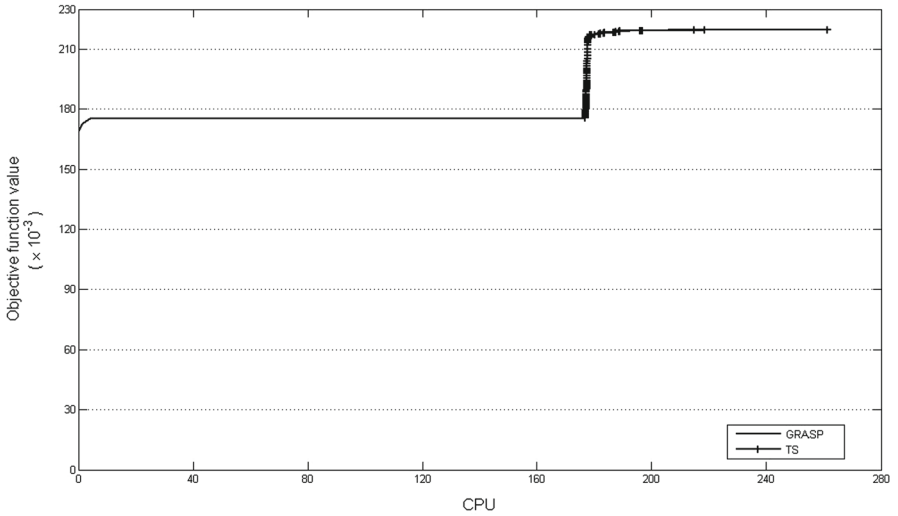

Fig. 2 Search profile of GRASP+TS on the RanReal240_01 instance

Table 9 Comparison of best methods on the 83 *MM* instances

Metric	PR_HMP (60 s)	PR_HMP (15 min)	AdTS_SO (60 s)	GRASP+TS (60 s)
Global average deviation	2.5 %	1.6 %	0.1 %	0.3 %
Number of best solutions	39	40	79	53
Score (worst = 249)	125	78	14	33
Time to best	19.2	256.6	42.9	14.8

In the third experiment, we compare the GRASP with Path Relinking by [13] for the handover minimization problem, PR_HMP, with the two best method identified in the previous experiments: AdTS_SO and GRASP+TS. GRASP with Path Relinking is run with the set of parameter values set in their experimentation [13]. It is worth mentioning that it employs a specific value for each parameter on each handover minimization instance. In our experiment, we apply these three methods to solve the 83 instances in the *MM* set. As in the previous experiments, we run each method for 60 s on each instance. However, since [13] reported very long running times, we have also run their PR_HMP method for 15 min. Table 9 shows the associated results in which we add to the metrics reported in the previous experiments, the running time to the best solution that each method is able to find (Time to best).

Results in Table 9 show that the three methods obtain high quality solutions on the handover minimization instances. The PR_MHP obtains slightly worst results than the other two methods. In fact, when this method is run for 15 min, it obtains a 1.6 % average deviation, while AdTS_SO and GRASP+TS obtain 0.1 and 0.3 % respectively running for 1 min. If we compare these two best methods, we see that AdTS_SO exhibits a marginal improvement over GRASP+TS but it requires a significant longer

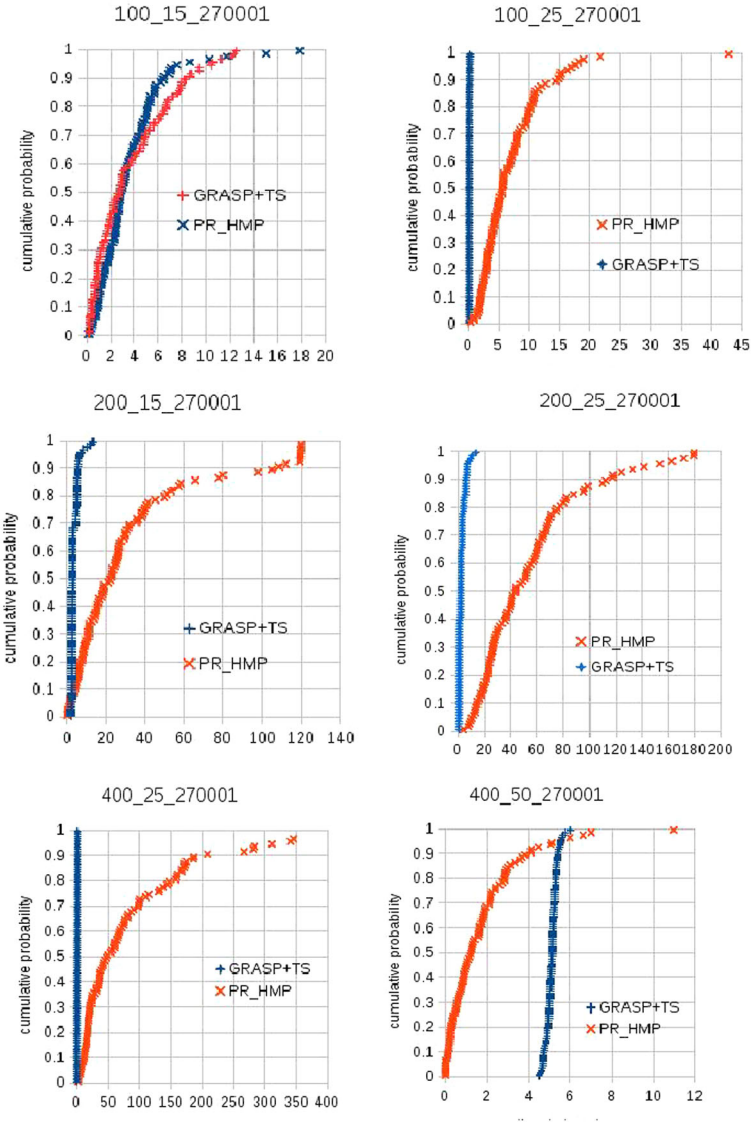

Fig. 3 Time to target plots

CPU time to reach the best solutions (42.9 s to find the best for AdTS_SO and 14.8 for GRASP + TS). We applied the Friedman test, and reject the hypothesis that there are no differences among the procedures with a p value lower than 0.001.

Aiex et al. [1] observed that the variable *time-to-target-value* in GRASP usually has an exponential distribution. Time-to-target (TTT) plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. TTT plots are used to

characterize the running times of stochastic algorithms for combinatorial optimization. Specifically, for each instance/target pair, the running times are sorted in increasing order. We associate with the i-th sorted running time t_i a probability $p_i = (i - 1/2)/n$ and plot the points (t_i, p_i). The resulting diagram shows the cumulative probability distribution plot and permits to check whether a given algorithm has or not an exponential distribution.

Considering that PR_HMP and GRASP+TS exhibit very similar times to best in Table 9, in our final experiment we plot their time to target values to compare their performance. In particular, we ran 100 times these two methods on six representative instances, stopping when a solution is found with objective value equal to the target for this instance. As recommended in [13], the target is obtained as the best known value multiplied by 1.02. For each run we recorded the running time. Each run is independent of the other by using a different initial seed for the random number generator (from 1 to 100 in each run respectively). With these 100 running times, we plot the 6 TTT plots (run time distributions) shown in Fig. 3. This experiment confirms that both methods are able to obtain high quality solutions in short running times. In most of the cases, GRASP+TS is faster than PR_HMP to reach the desired target. The plots in this figure also show the expected exponential runtime distribution for the two methods. Therefore, linear speed is expected if the algorithms are implemented in parallel.

6 Conclusions

The CCP is a difficult combinatorial optimization problem that is closely related to the MDGP, although this connection had not been previously discussed. Of particular interest in our work has been testing the effects of a variety of search strategies—such as those that allow the search to move outside the feasible region—and neighborhood structures within the GRASP framework. We also explored a tabu search design with a 2-1 exchange neighborhood that proved effective. Additionally, we have tested a GRASP with Path Relinking designed for a real application in the context of handover mobility networks, which turns out to be the minimization version of the CCP. The results of our experiments with 133 instances indicate that the proposed hybrid heuristic compares favorably to the existing procedures.

Acknowledgments This research has been partially supported by the University of Valencia (UV-INV-PRECOMP13-115334), *Ministerio de Educación Cultura y Deporte* of Spain (Grant Ref. PRX12/00016), the *Ministerio de Economía y Competitividad* of Spain (Grant Ref. TIN2012-35632) and the Generalitat Valenciana (Prometeo 2013/049). The authors would like to thank Profs. Morán-Mirabal, González-Velarde, Resende, and Silva for sharing their GRASP with PR code with them.

References

1. Aiex, R.M., Resende, M.G.C., Ribeiro, C.C.: TTT plots: a perl program to create time-to-target plots. Optim. Lett. **1**(4), 355–366 (2007)
2. Deng, Y., Bard, J.F.: A reactive GRASP with path relinking for the capacitated clustering. J. Heuristics **17**, 119–152 (2011)

3. Duarte, A., Sánchez-Oro, J., Resende, M., Glover, F., Martí, R.: Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization. Inform. Sci. **296**, 46–60 (2015)
4. Fan, Z.P., Chen, Y., Ma, J., Zeng, S.: A hybrid genetic algorithmic approach to the maximally diverse grouping problem. J. Oper. Res. Soc. **62**, 92–99 (2011)
5. Feo, T., Goldschmidt, O., Khellaf, M.: One-half approximation algorithms for the k-partition problem. Oper. Res. **40**, S170–S173 (1992)
6. Ferreira, C.E., Martin, A., de Souza, C.C., Weismantel, R., Wolsey, L.A.: The node capacitated graph partitioning problem: a computational study. Math. Program. **81**, 229–256 (1998)
7. Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP, Part I: Algorithms. Int. Trans. Operat. Res. **16**, 1–24 (2009)
8. Gallego, M., Duarte, A., Laguna, M., Martí, R.: Hybrid heuristics for the maximum diversity problem. Comput. Optim. Appl. **44**(3), 411 (2009)
9. Gallego, M., Laguna, M., Martí, R., Duarte, A.: Tabu search with strategic oscillation for the maximally diverse grouping problem. J. Oper. Res. Soc. **64**, 724–734 (2013)
10. Glover, F., Laguna, M.: Tabu Search. Kluwer, Boston (1997)
11. O'Brien, F.A., Mingers, J.: The Equitable Partitioning Problem: A Heuristic Algorithm Applied to the Allocation of University Student Accommodation. Research Paper No. 187. Warwick Business School, Coventry (1995)
12. Martí, R., Sandoya, F.: The equitable dispersion problem. Comput. Operat. Res. **40**, 3091–3099 (2013)
13. Morán-Mirabal, L.F., González-Velarde, J.L., Resende, M.G.C., Silva, R.M.A.: Randomized heuristics for handover minimization in mobility networks. J. Heuristics **19**, 845–880 (2013)
14. Ribeiro, C.C., Uchoa, E., Werneck, R.F.: A hybrid GRASP with perturbations for the Steiner problem in graphs. INFORMS J. Comput. **14**, 228–246 (2002)
15. Weitz, R.R., Jelassi, M.T.: Assigning students to groups: a multi-criteria decision support system approach. Decis. Sci. **23**(3), 746–757 (1992)
16. Weitz, R.R., Lakshminarayanan, S.: An empirical comparison of heuristic methods for creating maximally diverse groups. J. Oper. Res. Soc. **49**(6), 635–646 (1998)