GRASP with Evolutionary Path Relinking for the Conditional p-Dispersion Problem

Jesús Sánchez-Oro[a], Anna Martínez-Gavara[b,*], Ana D. López-Sánchez[c], Rafael Martí[b] and Abraham Duarte[a]

[a] Universidad Rey Juan Carlos, Dept. Computer Science and Statistics, C/Tulipan S/N, Móstoles, 28933, Madrid, Spain
[b] Universitat de València, Dept. d'Estadística i Investigació Operativa, C/Doctor Moliner, 50, Burjassot, 46100, Valencia, Spain
[c] Universidad Pablo Olavide, Dept. de Economía, Métodos Cuantitativos e Hist., Ctra. Utrera km 1, Sevilla, 41013, Sevilla, Spain

ARTICLE INFO

Keywords:
Combinatorial optimization
p-Dispersion
Conditional p-Dispersion
metaheuristics
GRASP
Path Relinking.

ABSTRACT

In this paper, we propose a new heuristic method that hybridizes GRASP with Path Relinking to solve the conditional p-Dispersion problem. Given n elements, from which $q < n$ have been already selected, this problem seeks to select $p < n$ additional unselected elements to maximize the minimum dissimilarity among them. The conditional p-dispersion problem models a facility location problem motivated by a real situation faced in many practical settings arising when some facilities have been already located. The algorithm includes a novel proposal based on an efficient interplay between search intensification and diversification provided by the Path Relinking component, and it also incorporates an intelligent way to measure the diversity among solutions. An extensive computational experimentation is carried out to compare the performance of our heuristic with the state of the art method. The comparison shows that our proposal is competitive with the existing method, since it is able to identify 17 best-known values. Additionally, our experimentation includes a real practical case solved for a Spanish company in its expansion process. This case illustrates both the applicability of the conditional p-dispersion model, and the suitability of our algorithm to efficiently solve practical instances.

1. Introduction

Maximum diversity problems consist in selecting a subset of elements from a given set in such a way that the diversity among them is maximized. These problems have been widely studied in the literature, and many methodologies have been applied to solve them. Martí et al. (2022) reviewed existing publications on diversity problems, and conducted a critical analysis to determine the best algorithms to solve them.

One of the most important models when dealing with diversity maximization is the p-dispersion problem (p-DP), also called the MaxMin dispersion problem (Kuby, 1987). In this model, the number of selected elements p is defined beforehand, and the diversity is measured as the minimum distance between each pair of elements in the subset P of selected elements ($|P| = p$). This problem is $\mathcal{NP}$-hard as proved by Erkut (1990). According to the geometrical study in Parreño et al. (2021), one of the main characteristics of the p-DP is that the selected elements are scattered and equidistant over the entire region, which makes this model well suited for location problems. This study recommends the p-DP as the best model for practical applications, especially in logistics.

In this paper, we target a variant of the p-DP proposed by Cherkesly and Contardo (2021), the conditional p-Dispersion Problem (c-pDP). As far as we know, this is the only publication devoted to this practical variant of the p-DP. The c-pDP consists in selecting p elements in such a way that the dispersion is maximized, but subject to a pre-established important constraint: q elements are already selected. The c-pDP has applications in location problems (Erkut and Neuman, 1989; Rahman and Kuby, 1995), portfolio optimization (Kudela, 2020), decision planner (Contardo, 2020), biological preservation (Glover et al., 1995, 1998), judiciary or committees evaluation (Adil and Ghosh, 2005; Weitz and Lakshminarayanan, 1998, 1997). In fact, many of the existing applications of the p-DP described in the literature are better modeled by the c-pDP. Consider, for example, the location problems in which we have to establish new hospitals or warehouses in a territory to provide a good level of service. It is more realistic

*Corresponding author
Email addresses: jesus.sanchezoro@urjc.es (J. Sánchez-Oro); gavara@uv.es (A. Martínez-Gavara); adlopsan@upo.es (A.D. López-Sánchez); rmarti@uv.es (R. Martí); abraham.duarte@urjc.es (A. Duarte)
ORCID(s): 0000-0003-1702-4941 (J. Sánchez-Oro); 0000-0001-9995-010X (A. Martínez-Gavara); 0000-0003-3022-3865 (A.D. López-Sánchez); 0000-0001-7265-823X (R. Martí); 0000-0002-4532-3124 (A. Duarte)

to consider that some of the facilities are already operating, and we have to consider them when locating the new ones, instead of assuming that we are locating all of them from scratch. The c-pDP models different applications, and its elements may represent facilities in location problems, investments in portfolio context, or group members in the selection of a committee. We include in the computational experimentation a case study from the food industry that we solved with the consulting company OGA (www.oga.ai), which triggered our interest on this problem.

This paper is not limited to simply solving the problem of the c-pDP, but it also proposes a methodology that can be applied to solve other combinatorial optimization problems. In particular, we propose a framework to apply PR in problems with Max-Min objective function such as the c-pDP (and it is aplicable to Min-Max problems as well). Specifically, exterior and interior PR strategies have been combined (hybridized) to deal with the flat landscape of the Max-Min objective function. It is well-documented in the optimization literature, that objective functions with the Min-Max or Max-Min types, provide very low information to guide the search of heuristic algorithms. Our method overcomes this lack of information with search indicators to guide the heuristic towards good regions in the search space. The combination between exterior and interior PR, coined as *Reactive Path Relinking* (Lozano-Osorio et al., 2023), was applied to solve the Bi-objective p-Median and p-Dispersion problem. To solve the c-pDP we apply the Reactive Path Relinking with diversification and intensification purposes by including different designs: the static, the dynamic, and the evolutionary GRASP with PR. Our computational study will disclose which variant fits better to c-pDP, the problem of interest in this paper.

To sum it up, the main contributions of this work are: (i) to compare different variants of GRASP with PR to solve c-pDP; (ii) to propose a GRASP construction phase that ensures diversity among all the constructed solutions; (iii) to combine exterior and interior path relinking strategies to allow diversification and intensification during the search; (iv) to propose a new metric to evaluate the diversity between different solutions in the context of max-min and min-max optimization problems; and (iv) to perform numerical experiments that reveal the best strategies for our problem, and to solve a real location application from the food industry.

The rest of this paper is organized as follows. In Section 2, we describe the mathematical model and we review the existing literature in c-pDP. Section 3 describes the GRASP with PR methodology, including its variants. Section 4 provides a detailed description of our proposal to solve the c-pDP, which constitutes a framework for other combinatorial optimization problems with objective functions of type Min-Max or Max-Min. Then, Section 5 reports on the experimental study, including a real application on a practical location problem. Finally, our conclusions are presented in Section 6.

2. Problem description and previous method

As mentioned in the introduction, the conditional p-dispersion problem, introduced by Cherkesly and Contardo (2021), is a realistic variant of the well-known p-dispersion problem. Given a set V of n elements, let $Q \subset V$ be a subset of $q < n$ elements already selected. The conditional p-dispersion problem (c-pDP) consists of selecting p new elements from the subset $V \setminus Q$ in such a way that the dispersion among the selected elements is maximized. Table 1 summarizes the nomenclature and mathematical symbols that we use in the paper to describe the model and the proposed algorithms.

The c-pDP can be formally defined as follows. Let $V = Q \cup R$ be a set of n elements where the elements in Q have been selected in advance, with $|Q| = q$. The elements in R have not been selected yet, with $R \cap Q = \emptyset$. Let D be the matrix with $D(i,j) = D(j,i) \geq 0$ that measures the dissimilarity between two elements $i, j \in V$ with $D(i,i) = 0$ for every $i \in V$. The goal of the c-pDP is to select a subset $P^* \subset R$ with $|P^*| = p$ that maximizes the overall minimum dissimilarity among all the selected elements (including the already selected (Q) and the new selected elements (P)). The c-pDP can be formulated as follows:

$$P^* = \underset{P \subset R : |P| = p}{\arg\max} \min_{i,j \in Q \cup P} D(i,j) \tag{1}$$

In a similar way that in the p-dispersion problem, the mathematical programming model for c-pDP is based on the binary variables x_i that take the value 1 if element i is selected and 0 otherwise. Then, it can be stated as follows:

Table 1
Symbols and Definitions.

Symbol	Definition
V	set of n elements
Q	set of q elements already selected from V, fixed in advance
R	complement set of Q: $V = Q \cup R$
P	feasible solution, set of the new p selected elements
$D(i, j)$	dissimilarity between element i and j
$f(P)$	objective function value of solution P

$$\begin{aligned} \max \quad & \min_{i,j \in Q \cup R : x_i = x_j = 1} D(i, j) \\ \text{s.t.} \quad & \sum_{i \in R} x_i = p \\ & x_i = 1 \qquad \forall\, i \in Q \\ & x_j \in \{0, 1\} \quad \forall\, j \in R. \end{aligned} \tag{2}$$

Let us define as $f(P)$ the objective function of the c-pDP, i.e., the minimum distance between each pair of elements in $P \cup Q$.

Cherkesly and Contardo (2021) propose an exact method to solve the c-pDP. Their proposal, called the exact decremental clustering algorithm, has the following main steps. Initially, it constructs a feasible solution using a heuristic algorithm. The objective function value of the constructed solution serves as a lower bound for the optimal solution. The algorithm then discards all elements with a distance to the fixed elements less than this lower bound ($\underline{z}$). It is clear that a better solution should contain elements with a distance larger than $\underline{z}$ from the selected elements. The algorithm takes advantage of this fact, reducing the size of the graph and, consequently, the computational time. Finally, the authors employ the exact algorithm proposed by Contardo (2020) for the p-DP on the reduced graph to identify a new solution or conclude that it does not exist. It is important to highlight that this proposal is executed within a prescribed time limit, which implies that not all instances are solved to optimality.

As shown in the computational experimentation of Cherkesly and Contardo (2021), their exact algorithm performs well on small instances, especially those with a small value of p. We have empirically found that its performance quickly deteriorates when increasing the value of p. This is to be expected since, as shown in Martí et al. (2022), the number of solutions exponentially grows with this parameter. Figure 1 illustrates this fact by showing the number of solutions in a small instance with $n = 25$ as a function of the number of selected elements m.

The main motivation of our paper is to target medium and large size instances, in line with the realistic applications of this problem, and to provide high-quality solutions to them.

3. GRASP with Path Relinking Methodology

Path Relinking (PR) is a trajectory-based metaheuristic mostly applied as an intensification strategy to enhance different optimization technologies. It was proposed in the context of Tabu Search (Glover and Laguna, 1997), and it has evolved into a more generic methodology, usually applied as a post-processing in GRASP (Resende et al., 2010).

The PR methodology explores trajectories that connect pairs of solutions (P_i, P_g), with the aim to find improved solutions along the path from the *initiating solution* P_i to the *guiding solution* P_g. The procedure generates *intermediate solutions* by combining attributes from both solutions, the *initiating* and the *guiding solutions*, iteratively including attributes of P_g into P_i until P_g is obtained.

Laguna and Martí (1999) incorporated PR into the framework of GRASP as a long-term intensification. In this context, the concept of relinking involves establishing a connection between a solution obtained through GRASP and a selected elite solution, also obtained with a previous GRASP iteration. As a result, the interpretation of relinking within GRASP differs from its original implementation in Tabu Search, where solutions from consecutive iterations are

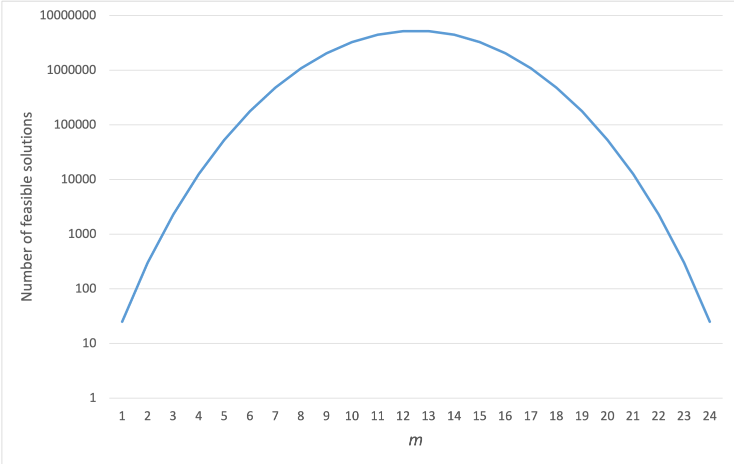

Figure 1: Number of solutions, as a function of $m = p + q$ in the x-axis, of an instance with $n = 25$.

linked through a sequence of moves. Moreover, this seminal work opened the door to other hybridizations in which PR is coupled with other heuristics, such as Variable Neighborhood Search, Iterated Local Search or Genetic Algorithms. These hybridizations resulted in improvements in solution quality and running times. According to Resende et al. (2010), there are several possibilities for combining GRASP with PR, resulting in different variants. In this paper we focus on the most successful ones: static, dynamic, and evolutionary PR.

3.1. Static Path Relinking

To simplify the description, and to keep it in generic terms, we consider that we have obtained an initial population, Pop, of N solutions ($|Pop| = N$), with the previous application of GRASP.

Static Path Relinking first selects a small set with the best GRASP solutions, and then operates on it (ignoring the rest of the solutions in the initial population). This set is called elite set (ES), and it contains the best solutions to be combined with PR. It must be noted that the meaning of best in this context refers to both quality and diversity. In particular, a subset of b solutions (where $b < N$) is selected from the initial population to form ES, and a solution P_j generated with GRASP is included in the ES if it satisfies any of the following conditions:

- The elite set is not entirely populated, i.e., $|ES| \leq b$.
- The elite set is complete, i.e., $|ES| = b$, and the solution P_j has better objective function value than the worst solution in ES. In this case, the most similar solution in ES to P_j is removed from ES (i.e., P_j replaces it).

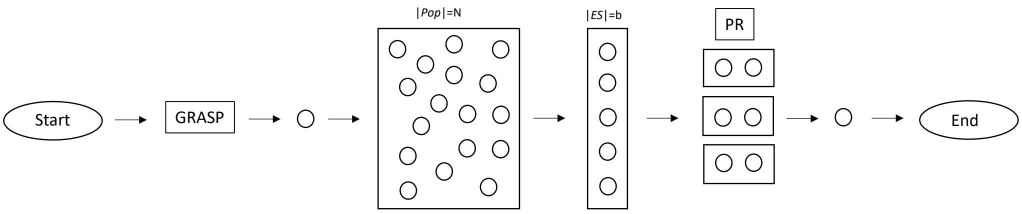

Figure 2: Static GRASP with PR.

Once the elite set with size b is created, PR is applied to combine every pair of solutions in ES. The flow diagram depicted in Figure 2 is referred to as the *static* approach because it involves two steps: initially applying GRASP to create the ES, and subsequently using PR to generate solutions by considering all pairs of solutions within ES. Note that these two steps are sequentially performed, and PR is applied when GRASP has already finished.

3.2. Dynamic Path Relinking

A different approach to implementing GRASP with PR involves a *dynamic* update of the ES, as in the original proposal by Laguna and Martí (1999). In this design, every solution generated using GRASP undergoes the PR algorithm directly. PR is applied between this solution and a solution randomly chosen from the ES.

It is worth mentioning that in this design the ES changes throughout the process. As depicted in Figure 3, GRASP constructs b solutions to initialize the ES. Subsequently, GRASP is executed, and PR is applied between each newly generated solution and a randomly selected solution from the ES. This process is repeated N times.

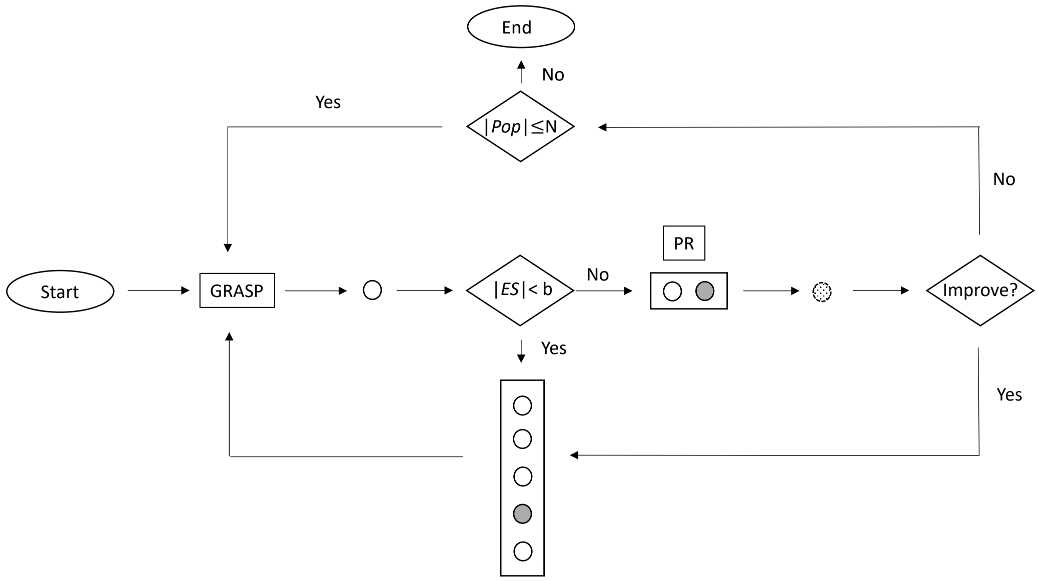

Figure 3: Dynamic GRASP with PR.

An important difference between the static and the dynamic designs is that, in the static PR, the solutions obtained from the application of PR are not considered to enter in the ES. In this variant, PR is applied as a post-processing phase to all the pairs of solutions in the ES; and after this, the method stops. On the other hand, in the dynamic design, the solutions obtained with PR are considered to become part of ES. Therefore, it is possible to replace a solution in the elite set if any of the intermediate solutions qualify. Consequently, iterations in this design have two steps, the first one applying GRASP, and the second one PR, and the method performs iterations until the stopping criterion is reached.

3.3. Evolutionary Path Relinking

This is the most complex PR implementation proposed so far, since it combines the dynamic and static variants in an effective way. We usually refer as *GRASP with EvPR* to its hybridization with GRASP proposed first by Resende and Werneck (2004).

In the first stage, GRASP with EvPR follows a similar approach that the dynamic strategy. In each iteration, it applies both GRASP, as well as PR, to obtain the elite set (see Figure 4). After a predetermined number of iterations, the first stage terminates. This is depicted in the bottom part of the figure highlighted in gray.

In the second stage, GRASP with EvPR applies a post-processing phase based on the static design. In particular, PR is applied to every pair of solutions in the ES. The solutions obtained through this subsequent application of PR are considered as potential candidates to enter in the ES, thus replacing the elements there. This process allows the elite set to evolve over time, and this is why the method is called Evolutionary PR. Solutions in the ES are combined (i.e., submitted to PR) until the time limit is reached or until no improvement is attained from one iteration to the next one. This second stage is represented in the upper right part of the diagram in Figure 4.

GRASP with EvPR is based on the evolution of a small set of selected solutions (Resende and Werneck, 2004). Algorithm 1 shows the pseudo-code for the GRASP with EvPR. The inputs of the algorithm are the parameters α and

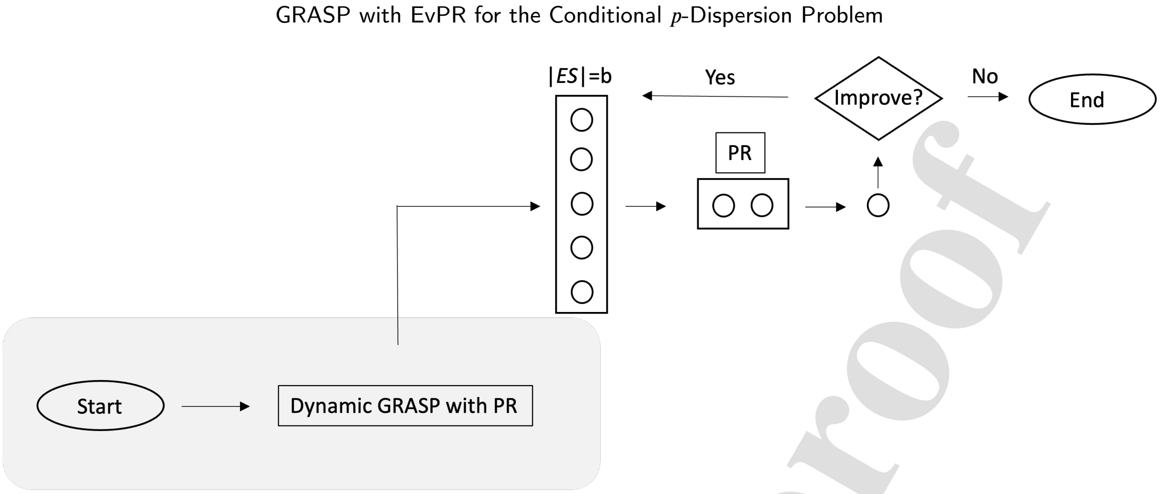

Figure 4: GRASP with Evolutionary PR.

b. Parameter α balances the greediness and randomness of GRASP, and parameter b is the size of the elite set. In the first while-loop (from step 3 to 18), the pseudo-code of the dynamic GRASP with PR is shown. First, the algorithm populates the set ES using GRASP (steps 4 to 8). Each new incumbent solution P (steps 9 and 10) is hybridized with a randomly selected solution from the elite set. Then, the best solution found during the path is incorporated in the ES if an intermediate solution has a better objective function value, and the set is updated. The specific details of the algorithm for solving the conditional p-dispersion is defined in Section 4.

4. The proposed algorithm for the Conditional p-Dispersion

This section focuses on applying the three variants of GRASP with PR described in Section 3 to solve the Conditional p-Dispersion problem. Additionally, we propose the use of two different PR strategies, interior PR (IPR) and exterior PR (EPR) in order to provide intensification and diversification in the search, respectively.

Although path relinking has been applied to many optimization problems, most of these implementations limit themselves to the IPR, which creates a path between two solutions. We may consider that this IPR somehow generalizes the concept of the so-called convex combination in global optimization as long as it explores the solutions that rely between two given solutions. In this paper however, we complement this exploration with the beyond-form of path relinking (non-convex exploration) proposed in Glover (2014) and called *Exterior Path Relinking*. Instead of introducing into the initiating solution elements from the guiding solution, EPR introduces in the initiating solution elements not present in the guiding solution. In this way, we may say that EPR separates or disconnects two elite solutions (Lozano-Osorio et al., 2023).

In this section, we first describe our heuristic algorithm for solving the c-pDP. We introduce the search elements and key characteristics of the proposed GRASP for c-pDP (see Subsection 4.1), and then, we describe our Path Relinking implementation for this problem (4.2). Note that we propose several search strategies that include not only those mentioned above, but also some reactive mechanisms that permit our heuristic to automatically adapt itself to efficiently solve each specific instance.

4.1. GRASP

As briefly introduced, the PR strategy begins with the creation of an elite set consisting of high-quality and diverse solutions. One of the features of the Greedy Randomized Adaptive Search Procedure (GRASP) is its ability to balance the quality and the diversity of the generated solutions. In general terms, GRASP (Feo et al., 1994) is a multi-start algorithm in which each iteration consists of two phases: construction phase and improvement phase. As its very name indicates, the construction phase builds initial feasible solutions using a greedy randomized adaptive algorithm and the improvement phase applies a local search by exploring the neighborhood solutions in order to escape from possible local optimum. Usually, the output of the GRASP is the best solution found along a number of N iterations. However, when combining GRASP with PR, a subset of GRASP solutions are kept in the elite set to apply the PR post-processing.

Algorithm 1 `GRASP with Evolutionary PR(α,b)`

```
f* ← −∞                                                   ▷ Initialize the objective function value
ES ← ∅                                                    ▷ Initialize the Elite Set
while stopping criterion is not satisfied do
    while |ES| < b do
        P ← Greedy_Randomized_Adaptive_Construction(α)    ▷ see Section 4.1.1
        P ← Local_Search(P)                               ▷ see Section 4.1.2
        ES ← {P}
    end while
    P ← Greedy_Randomized_Adaptive_Construction(α)        ▷ see Section 4.1.1
    P ← Local_Search(P)                                   ▷ see Section 4.1.2
    P′ ← Select_Solution(ES)
    P″ ← PR(P′, P)                                        ▷ see Section 4.2
    ES ← Update_Elite(ES, P″)
    if f(P″) > f* then
        P* ← P″
        f* ← f(P*)
    end if
end while
while ∃ P_i, P_g ∈ ES not yet relinked do
    P ← PR(P_i, P_g)                                      ▷ see Section 4.2
    ES ← Update_Elite(ES, P)
    if f(P) > f* then
        P* ← P
        f* ← f(P*)
    end if
end while

return P*
```

4.1.1. Construction phase

Solutions for the c-pDP consist of subsets $P \subset R$ of p selected elements. To implement the GRASP construction, a Candidate List (CL) is created containing all available elements to be added to the solution (initially, $CL = R$). Subsequently, a subset of the CL, known as the Restricted Candidate List (RCL) is calculated containing the most promising candidate elements to be added to the solution. An element is considered if its greedy function value exceeds a threshold τ, which is calculated as: $\tau = g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$, where the parameter α is responsible for controlling the elements that will be part of the RCL and $g_{\min}$ and $g_{\max}$ are the minimum and maximum value of the greedy function $g(i) = \min_{j \in P} D(i, j)$ for all candidate elements $i \in CL$, respectively. Note that, if $\alpha = 1$, the RCL will contain all candidate elements, so RCL and CL are equal and therefore, the construction is totally random. On the contrary, if $\alpha = 0$, the RCL will only contain the most promising element to be added to the solution, therefore the construction is totally deterministic. An element is randomly selected from the RCL to be included in the solution under construction P, and then the CL is updated by removing the selected element. This process is repeated until a feasible solution is reached.

To address the c-pDP, the greedy function is directly the objective function of the problem. Therefore, the most promising element to be added to the solution will be the furthest from all already selected elements (see equation 1). An important strategy in the method is that, as previously mentioned, solutions need to be both good and diverse. By generating solutions based on the increment of the objective function value, we ensure their relative quality. However, to achieve diversity, the algorithm checks that each newly generated solution differs from previously constructed ones. If the new solution shares the same elements as any previously generated solution, it is discarded as it has already been examined.

4.1.2. Improvement phase

Once a solution has been constructed, the improvement phase is applied. During this phase, the solution is enhanced through the execution of a standard local search until a termination criterion is satisfied. We define $N(P)$ as the neighborhood of solution P, which is determined by swap moves. Specifically, $N(P)$ consists of the set of all solutions that can be reached by exchanging an element $u \in P$ with an element $v \in R \setminus (P \cup Q)$. In mathematical terms:

$$N(P) = \{P' \subset V : P' = P \setminus \{u\} \cup \{v\}, u \in P, v \in R \setminus (P \cup Q)\} \tag{3}$$

The proposed local search instead of performing a complete exploration of the search space only considers strategic exchanges not only to save computing time, but also because we are only interested in the movements able to reduce the value of the objective function. This local search is based on similar local search strategies have been proposed in related problems (Lozano-Osorio et al., 2022; Lu et al., 2023).

Our improvement method creates two lists and conducts exchanges between elements in both lists. On the one hand, `in_list` contains the elements of the solution with the minimum distance value, i.e., `in_list` $= \{i \in P : d_i = d^*\}$, where $d_i = min_{j \in Q \cup P} D(i, j)$ and d^* is the objective function value of the solution P, $d^* = min_{i,j \in Q \cup P} D(i, j)$. On the other hand, `out_list` contains the non-selected elements with distances larger than d^*, i.e., `out_list` $= \{j \in R \setminus P \cup Q : d_j > d^*\}$. To improve the objective function value, the algorithm exchanges elements in `in_list` with elements in `out_list`.

4.2. Reactive Path Relinking

In this work, we propose to hybridize two PR methodologies, Interior and Exterior Path Relinking instead of applying a standard PR. This algorithm has been recently proposed by Lozano-Osorio et al. (2023), and it is coined as Reactive Path Relinking (RPR). The rationale behind the RPR is to intensify when two solutions are different (by applying IPR) and diversify when two solutions are similar (by applying EPR). IPR and EPR are explained in detail in the following subsections.

4.2.1. Interior Path Relinking

IPR creates a path that connects two good solutions, exploring new intermediate solutions attaining a high probability of finding good solutions in the path that connects them. This is the standard PR. This strategy intensifies the search by exploring repeatedly promising regions of the search space.

The path between P_i and P_g is created by including in P_i elements that are in $P_g \setminus P_i$, exchanging them with those elements in $P_i \setminus P_g$. Then, at every step, the IPR modifies the current intermediate solution P_i^j to become more similar to P_g, until it is reached.

Figure 5 shows an example of the IPR where the initiating and guiding solutions are $P_i = \{1, 2, 3, 4, 5, 6\}$ and $P_g = \{1, 2, 3, 7, 8, 9\}$, respectively. At the first step of the IPR, element 4 is replaced by element 7, obtaining the intermediate solution $P_i^1 = \{1, 2, 3, 7, 5, 6\}$, then, at the second step, element 5 is interchanged by element 8, resulting in other intermediate solution, $P_i^2 = \{1, 2, 3, 7, 8, 6\}$, and at the last step, element 6 is substituted by element 9 obtaining, in such a way, the guiding solution, P_g. Similarly, to transform P_g into P_i, the first step of the IPR replaces element 7 by element 4, obtaining the intermediate solution $P_g^1 = \{1, 2, 3, 4, 8, 9\}$, then, at the second step, element 8 is interchanged by element 5, resulting in the solution $P_g^2 = \{1, 2, 3, 4, 5, 9\}$, and the last step substitutes element 9 by element 6 obtaining, in such a way, the initiating solution, P_i.

Notice that the IPR is applied when the two high-quality solutions are sufficiently different according to a given distance metric. In Section 4.2.3, we propose a new metric and compare it with the conventional one.

4.2.2. Exterior Path Relinking

EPR follows the opposite idea of IPR in geometrical terms and creates a path that "disconnect" or "separates" two promising solutions given that they are similar, leading to explore new solutions. This strategy inputs diversity in the search by exploring different regions of the search space.

Given an initiating solution P_i and a guiding solution P_g, EPR iteratively includes in P_i elements that are not in P_g, with the aim of reaching new solutions which are diverse with respect to both P_i and P_g.

Figure 6 shows an example of the EPR where $P_i = \{1, 2, 3, 4, 5, 6\}$ and $P_g = \{1, 2, 3, 7, 8, 9\}$ are two pairs of solutions. The task is to eliminate the shared elements, $P_i \cap P_g = \{1, 2, 3\}$, from both solutions. At the first step of the EPR, element 1 is replaced with element 10, getting the intermediate solution $P_i^1 = \{10, 2, 3, 4, 5, 6\}$. At the

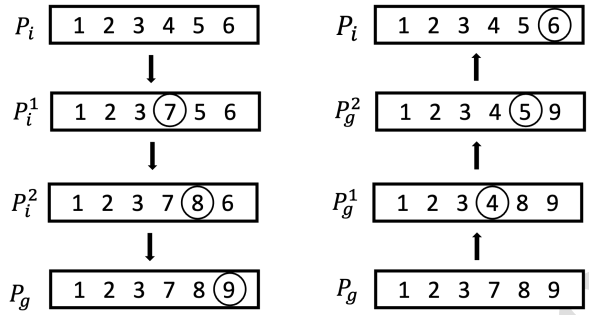

Figure 5: Interior Path Relinking.

second step, element 2 is interchanged by element 11, obtaining $P_i^2 = \{10, 11, 3, 4, 5, 6\}$. At the third step, element 3 is substituted by element 12, resulting in the intermediate solution $P_i^3 = \{10, 11, 12, 4, 5, 6\}$. Similarly, if we start with the solution P_g thought P_i, $P_g = \{1, 2, 3, 7, 8, 9\}$, then the first step of the EPR interchanges element 1 with element 10, resulting in the intermediate solution $P_g^1 = \{10, 2, 3, 7, 8, 9\}$. The second step of the EPR replaces element 2 by element 11, getting the solution $P_g^2 = \{10, 11, 3, 7, 8, 9\}$. And finally, at the last step of the EPR, the element 3 is substituted by element 12, obtaining the intermediate solution $P_g^3 = \{10, 11, 12, 7, 8, 9\}$.

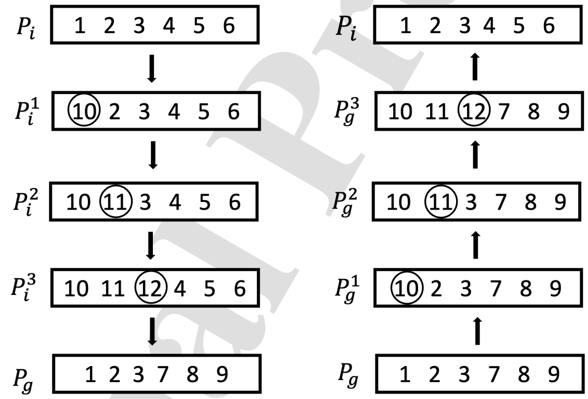

Figure 6: Exterior Path Relinking.

Notice that the EPR is applied when two solutions contain similar elements to diversify the search (see Section 4.2.3).

4.2.3. Evaluating solution diversity

To measure the *diversity* between two solutions P_i and P_j, a standard metric γ is the number of different elements between both solutions, i.e., $\gamma(P_i, P_j) = n - |P_i \cap P_j|$. Therefore, the diversity between a given solution P_j and the set of solutions in ES is measured as the distance to the nearest solution P_j in the elite set ES. Formally, $\gamma(P_j, ES) = \min_{P_i \in ES}\{n - |P_j \cap P_i|\}$.

Let $P_a = \{1, 2, 3, 4, 5, 6\}$, $P_b = \{1, 2, 3, 4, 7, 8\}$, and $P_c = \{1, 2, 7, 8, 9, 10\}$, be three feasible solutions, then $\gamma(P_a, P_b) = 2$, since they have two different elements, and $\gamma(P_a, P_c) = 4$, since they have four different elements. Therefore, solution P_a is more diverse compared to P_c than compared to P_b.

In this problem, given that the objective function value is determined by the shortest edge, we propose an alternative metric to evaluate the diversity between two solutions P_i and P_j. This new *diversity metric*, denoted by δ, takes the value 1 if the shortest edge of solution P_i is different to the shortest edge of solution P_j and 0, otherwise. Therefore, the diversity of a solution P_j compared to the ES solutions is 1 if the shortest edge of P_j differs from the shortest edges in all solutions in ES. The diversity $\delta(P_j, ES)$ is calculated for each solution $P_j \in ES$. The idea is to add a solution to ES if the value of the δ diversity is 1. Note that this way of measuring the diversity is valid not only for the conditional p-dispersion problem but also for the optimization problems in which the objective function is focused on maximizing a minimum or minimizing a maximum in which case, it is necessary to define δ as 1 if the largest value of solution two solutions are different.

Let $P_a = \{1, 2, 3, 4, 5, 6\}$, $P_b = \{1, 2, 3, 4, 7, 8\}$, and $P_c = \{1, 2, 7, 8, 9, 10\}$, be three feasible solutions. Furthermore, let's assume that $(1, 2)$, $(1, 2)$ and $(7, 10)$ are the shortest edges of P_a, P_b and P_c, respectively. Then $\delta(P_a, P_b) = 0$, since they have the same shortest edge so both solutions are similar, but $\delta(P_a, P_c) = 1$ or $\delta(P_b, P_c) = 1$, since they do not have the same shortest edge so such solutions are considered diverse.

At this point, it is worth mentioning how the diversity metric is used in the RPR. As previously mentioned, RPR applies IRP if two solutions are different, that is, if they are diverse, meanwhile RPR applies EPR if two solutions are similar. To determine which of the PR variants must be used by the RPR, it is crucial to know how the diversity will be measure.

Note that if we use γ to measure the diversity, it is considered that two solutions are different if they have more than a half elements different, and so, IPR is applied, otherwise, EPR is applied. When measuring the diversity with δ, it is much more easier since the value 1 indicates that solutions are different, so IPR should be applied, otherwise, value 0 indicates that solutions are similar, so EPR should be applied.

4.3. Computational complexity

This section is devoted to present the computational complexity of the proposed algorithm. In particular, each component of the algorithm is first independently analyzed and, then, the complexity of the complete proposal is computed. The complexity is analyzed in terms of the Big O notation, which refers to the worst case scenario to establish the bounds of the algorithm.

The first component of the algorithm is the constructive procedure presented in Section 4.1.1. The creation of the CL has a complexity of $O(n)$, since it requires to add every single candidate to the list. Then, in each iteration, the RCL is computed with a complexity of $O(n)$ and, after selecting the next element, the remaining ones in CL are updated, with a complexity of $O(n)$. Therefore, the final complexity of each iteration is $\max(O(n), O(n)) = O(n)$. Then, the complete constructive procedure has a complexity of $O(n^2)$, since it has a complexity of $O(n)$ in each iteration, requiring to perform n iterations, $(O(n \cdot n) = O(n^2))$.

The second component evaluated is the local search method described in Section 4.1.2. This is usually the most computationally demanding part of any optimization algorithm, so it has to be carefully implemented in order to have an efficient procedure. In each iteration of the local search, all the elements of the solution are examined in the worst case and, for each iteration, the evaluation of the movement is performed with a complexity of $O(n)$. Therefore, the complete complexity of the local search procedure is $O(n^2)$.

Finally, the complexity of the complete Evolutionary Path Relinking procedure is evaluated. The first stage of the algorithm consists of generating the initial elite set. This stage creates a fixed number of solutions and the complexity of each generated solution is the maximum between the complexity of the constructive procedure and the complexity of the local search method. Therefore, generating n solutions results in a complexity of $O(n \cdot n^2) = O(n^3)$. The second stage is the Dynamic PR. In this phase, for each iteration a solution is constructed and improved, resulting in a complexity of $O(n^2)$. Then, the combination method is applied. The proposed combination methods traverses all the candidates and perform a local search to each solution in the path, with a total complexity of $O(n^3)$. Hence, the complexity of Dynamic PR is evaluated as $\max(O(n^2), O(n^2), O(n^3)) = O(n^3)$.

The final stage of the algorithm is the evolution of the elite set applying path relinking. In this case, the procedure iterates while an improvement is found, with linear complexity. In each iteration, it is required to consider every pair of solutions to be combined, with a complexity of $O(n^2)$, performing a combination over every pair of solutions with a complexity of $O(n^2)$. Thus, the complete complexity of this phase is equal to $O(n^4)$.

Summarizing, the final complexity of the proposed algorithm is the maximum among all the evaluated elements, resulting in $\max(O(n^2), O(n^2), O(n^3), O(n^4)) = O(n^4)$. It is interesting to remark that this evaluation refers to the worst-case scenario, which is not the most common situation of the algorithm. Therefore, as it is customary in "heuristic papers", to have a more precise and representative analysis of the performance of the method, we perform an experimental evaluation via simulation over a set of representative instances, which permits to draw significant conclusions, as described in the following in Section 5.

Table 2
Comparison among different values for α parameter in GRASP.

α	T(s)	OF	Dev(%)	#Best
RND	0.10	12251.29	0.94	73
0.1	0.07	12047.88	1.63	55
0.2	0.08	12064.61	1.63	48
0.3	0.08	12065.79	1.48	56
0.4	0.09	12155.39	1.33	64
0.5	0.10	12167.34	1.18	69
0.6	0.11	12154.90	1.23	66
0.7	0.11	12095.57	1.09	68
0.8	0.12	12225.22	0.80	77
0.9	0.13	12247.71	0.60	94

5. Computational experiments

This section is devoted to present and discuss the computational experiments. We use the same set of 40 instances solved by Cherkesly and Contardo (2021) taken from the well-known TSPLIB (Reinelt, 1991), with a number of elements ranging between 1621 and 104815. To adapt these instances to the conditional p-dispersion problem, the authors generated the set Q using three strategies: greedy, optimal and random, and with the values of q and p being 5, 10, 15 and 20, obtaining a total of 16 combinations. In this work, we consider the set Q generated using the same greedy algorithm. For further details about the generation of the instances see Cherkesly and Contardo (2021). Therefore, a total of 640 instances are solved to perform a fair comparison with the state-of-the-art. All instances were solved on an AMD Ryzen 5950x with 128 GB RAM, and the algorithms were implemented using Java 17. Note that throughout this section, tables summarize the results obtained in each specific experiment. All instances, source code, and detailed results are publicly available at `https://grafo.etsii.urjc.es/cpDP`.

To evaluate the performance of each procedure, we consider the following metrics:

- $T(s)$, the total time in seconds.
- OF, the average of the objective function value.
- $Dev(\%)$, average value of the percentage deviation with respect to the best solution found in the experiment.
- $\#Best$, the number of best solutions with each algorithm.

Results are divided into two subsections: preliminary experiments and final results. In the preliminary experiments, parameters are tuned and furthermore, the contribution of each component of the algorithm is tested. This section is also called "scientific testing" in the literature on optimization to highlight that it provides insight on the strategies and elements of the algorithm. To avoid overfitting, we have selected a subset of 25% instances, in total 160 instances. As customary in the literature, in the preliminary experiments, GRASP was performed 100 iterations, and the size of the ES is fixed at 10 solutions. Once the final configuration of our proposal is set, we present a comparison against the state-of-the-art in the final results, also called competitive testing.

5.1. Preliminary experiments

As previously explained our proposal requires an initial population of solutions that is built with GRASP. Therefore, the first preliminary experiment is focused on selecting the best parameter α for the GRASP algorithm. To that end, we have generated 100 solutions by testing values ranging from 0.1 to 0.9 with a step of 0.1. Additionally, the value RND indicates that a random value for α is selected in each construction. Table 2 shows the results obtained in this experiment.

As can be drawn from the results, the quality of the solutions increases with the value of α. This indicates that diversity plays an important role in the proposed algorithm. Although the best results are found by both $\alpha = 0.9$ and $\alpha = RND$, we select $\alpha = RND$ for the remaining experiments since it provides equivalent results but with more diversity.

Table 3
Comparison between two diversity strategies in Static GRASP with RPR.

	T(s)	OF	Dev(%)	#Best
δ-RPR	0.78	12258.35	0.04	152
γ-RPR	0.77	12257.17	0.12	149

Table 4
Comparison among GRASP and the different GRASP with PR strategies proposed.

	T(s)	OF	Dev(%)	#Best
GRASP	1.14	12197.01	1.30	73
Static GRASP with RPR	1.30	12245.00	0.96	92
Dynamic GRASP with RPR	1.46	12300.24	0.70	105
GRASP with EvRPR	2.52	12363.47	0.23	131

Given the nature of the c-pDP, where the objective function is a max-min mathematical expression (see Section 4.2.3), we have introduced a new diversity metric, δ, to compare two solutions in the RPR, instead of the traditional diversity metric, γ. Therefore, the following experiment evaluates the relevance of selecting an adequate diversity metric. Moreover, we analyze the impact of the proposed diversity metric in the context of RPR.

The two strategies compared are the static GRASP with RPR with the δ diversity metric, which considers the diversity between two solutions with respect to their objective function value, and the traditional strategy, based on the γ diversity metric, which evaluates the number of different elements selected in each solution. We called them as δ-RPR and γ-RPR, respectively. Table 3 shows the results obtained in this experiment performed using the standard parameters for Pop and ES sizes, which are 100 and 10, respectively.

Although both strategies present similar results in terms of quality and computing time, it can be seen that the new diversity metric performs slightly better, being able to reach a larger number of best solutions with a deviation which is almost 0. Therefore, we select δ-RPR as the best Static GRASP with RPR strategy.

At this point, we intend to compare the Static GRASP with PR, the Dynamic GRASP with PR and the EvPR where the variant of PR is a δ-RPR as in the previous variants. The objective of this comparison is to evaluate which is the best proposal to solve the considered problem. Table 4 shows the results when comparing GRASP, GRASP with Static RPR, GRASP with Dynamic RPR, and GRASP with EvRPR.

Table 4 provides relevant information on the performance of the algorithmic variants proposed. First, it is important to remark that GRASP isolated is able to provide high-quality solutions, being competitive with the PR variants. However, these results suggest that applying PR increases the portion of the search space explored, thus leading to better results. Specifically, the best PR variant is clearly EvPR, which is able to reach 131 out of 160 best solutions. Furthermore, the deviation of 0.23% indicates that, in those instances in which EvPR is not able to reach the best solution, it still remains really close to it, emerging as the best PR strategy for the c-pDP. Regarding Static and Dynamic PR, the dynamic nature of the latter is a key feature to find better results than the former. Although both present high-quality solutions, Dynamic PR is slightly better, with a smaller deviation and a larger number of best solutions found. Finally, the computing time for all the variants is negligible, requiring, on average, approximately 2 seconds, being GRASP with EvRPR the slowest variant. Overall, we consider GRASP with EvRPR as our reference method, to be applied in comparison with all instances in the Competitive-experiments subsection.

The final preliminary experiment is designed to determine the best values for the population and elite set size. The main objective is to evaluate if the combination of these parameters contributes to an improvement of the average objective function value without a significant increase in the required computing time. Specifically, we have combined both parameters in the following way: $N = 100$ with $|ES| = 10$; $N = 250$ with $|ES| = 10$ and 25; and finally, $N = 500$ with $|ES| = 10$, 25, and 50. In this manner, the maximum size of the elite set is limited to 10% of the population size to keep a reasonable proportion in this subset of good and diverse solutions compared to the initial population.

As shown in Table 5, the larger the population size, the better the average objective function value. Evidently, the sample size of the solution space increases with larger population sizes, leading to higher solution quality.

Table 5
Parameter comparison in GRASP with EvRPR.

N	$\|ES\|$	T(s)	OF	Dev(%)	#Best
100	10	8.23	12497.82	0.33	114
250	10	16.55	12482.09	0.28	125
	25	35.68	12491.89	0.20	127
500	10	39.59	12500.78	0.10	141
	25	53.44	12506.79	0.12	142
	50	122.48	12507.66	0.05	150

Consequently, this increase results in longer execution times. Moreover, the improvement is not as significant when the size of the elite set increases. In conclusion, the combination parameter $N = 250$ and $|ES| = 25$ offers the best trade-off between solution quality and computational time.

5.2. Competitive experiments

In the previous section, the configuration of the algorithmic proposal has been set. Specifically, the GRASP parameter α is randomly selected at each iteration of the algorithm see Table 2. The diversity metric in RPR is the metric δ, as indicated in Table 3. Additionally, we use the best variant among the four considered in Table 4, that is, the GRASP with EvRPR, with $N = 250$ and $|ES| = 25$ (see Table 5). Therefore, in this section, we focus on a comparison with the state-of-the-art algorithm (SOTA). Specifically, we compare GRASP with EvRPR, and the SOTA on the testbed set, which consists of 640 instances with values of p: 5, 10, 15, and 20.

In this first experiment, we test the ability of GRASP with EvRPR to match the best-known values obtained by the SOTA algorithm (Cherkesly and Contardo, 2021). We run our heuristic once on each instance, and then, we summarize the comparative between SOTA and GRASP with EvRPR using statistical metrics in Table 6. Detailed results are available in the aforementioned repository.

It is well known that a fair empirical comparison between two algorithms relies on running them under the same conditions. This mainly implies to run them on the same computer and for the same time. Although the authors of the previous heuristic, SOTA, kindly shared their implementation with us, their executable program does not permit to adjust the running time, which makes a comparison with other methods complicated. In our experiments with their SOTA program, we observed a large variability in its running times across instances. We therefore classified the results according to these running times, that range from 10 to 10000 seconds in three groups (below 60 seconds, between 60 and 600 seconds, and above 600 seconds).

It is well accepted in the optimization literature that heuristics are meant to be fast. In line with that, we design our method, GRASP with EvRPR, to obtain high-quality solutions in short computational times (i.e., in few seconds in most of the cases). We then compare our method with the previous heuristic in the instances in which its program returns a solution in short running times (say lower than 60 seconds). For the sake of completeness, we include in Table 6 the summary results of both methods (SOTA and GRASP with EvRPR) in all the instances, although we do not believe that their comparison when the running times are so different is adequate.

Table 6 shows that GRASP with EvRPR outperforms SOTA in terms of average running times, percent deviation, and objective function value on the instances where SOTA is able to produce a solution in requires less than 60 seconds of CPU time. The average deviation from the best-known solution is 0.07% and our proposal achieves the best value in 345 out of 367 instances. For the instances where SOTA requires more than 60 seconds, it exhibits slightly better metrics than our algorithm. However, SOTA requires 13 times more CPU time than our algorithm (199.7 seconds versus 15.5 seconds) for the instances with execution times between 60 seconds and 600 seconds, and more than 24 times for the instances with more than 600 seconds of running time. To complement this analysis, we performed a paired-samples t-test[1] (see Demšar (2006) for more details on the statistical test). This test resulted in a $p - value$ of 0.0145, which is not less than the significance level of 0.01, and there is no significant differences between GRASP with EvRPR and SOTA when grouping all the instances together. Note that this is remarkable considering the extremely short running times of our algorithm.

[1] We chose this test because with a sample size of at least 40, the Wilcoxon W statistic tends to follow a normal distribution.

Table 6
Comparison of SOTA with GRASP with EvRPR on testbed instances with $p \leq 20$.

SOTA with		SOTA				GRASP with EvRPR			
CPU time	p	T(s)	OF	Dev(%)	#Best	T(s)	OF	Dev(%)	#Best
below 60s.	5	21.6	13510.5	0.66	131	0.1	13521.1	0.00	136
(367 inst.)	10	26.3	6250.4	0.39	99	0.6	6255.1	0.00	101
	15	31.0	5488.8	0.24	75	1.7	5490.6	0.04	69
	20	31.3	5034.9	0.00	51	2.4	5031.6	0.22	39
	all	27.5	7571.1	0.32	356	1.2	7574.6	0.07	345
between 60s.	5	168.1	31853.9	0.79	13	0.5	31874.1	0.00	14
and 600s.	10	205.5	26249.2	0.00	43	4.9	26226.6	0.37	34
(169 inst.)	15	217.9	13483.5	0.35	50	18.5	13483.5	0.31	44
	20	207.2	7200.7	0.21	57	38.2	7161.5	0.69	32
	all	199.7	19696.8	0.34	163	15.5	19686.4	0.34	124
above 600 s.	5	930.9	106107.6	0.00	5	3.5	106107.6	0.00	5
(88 inst.)	10	3015.7	67205.9	0.00	10	61.9	67172.8	0.33	8
	15	3516.7	39414.6	0.00	27	167.1	39396.9	0.76	17
	20	7756.7	28919.6	0.00	46	405.8	28655.0	1.39	16
	all	3805.0	60411.9	0.00	88	159.6	60333.1	0.62	46

0.00 means less than 0.001

Table 7
New best-known objective function values.

Instance	p	q	SOTA algorithm	GRASP with EvRPR
brd14051	5	10	1554	1780
brd14051	5	15	1264	1460
brd14051	10	15	1125	1194
brd14051	15	15	1040	1079
brd14051	5	20	1082	1460
brd14051	10	20	969	1186
brd14051	15	20	880	1079
brd14051	20	20	838	872
d18512	5	5	2258	2540
d18512	5	10	1758	2000
d18512	5	15	1419	1616
d18512	10	15	1261	1323
d18512	15	15	1194	1219
d18512	5	20	1250	1460
d18512	10	20	1155	1305
d18512	15	20	1041	1194
d18512	20	20	1000	1091

We now analyze the instances in which our method surpasses the previous one. In particular, GRASP with EvPR algorithm outperforms the best-known value found so far in 17 instances. Specifically, in Table 7, we provide the names and objective function values for these 17 instances in both algorithms.

To complement the above findings, we evaluate the performance of GRASP with EvRPR across a total of 220 instances. We combine parameter values of q (5, 10, 15, and 20) with larger values of p (25 and 30) while maintaining similar computational times. Table 8 shows that our proposed method achieves an average percent deviation of 0.05% from the best-known solutions and it finds the best solution for 218 out of 220 instances. In contrast, the state-of-the-art (SOTA) method finds only 99 best solutions with an average deviation of 0.65%. There are statistical significant

Table 8
Comparison of SOTA with GRASP with EvRPR on testbed instances with $p = 25, 30$.

	SOTA			GRASP with EvRPR		
p	OF	Dev(%)	#Best	OF	Dev(%)	#Best
25	7308.41	0.49	58	7339.98	0.07	113
30	6328.50	0.83	41	6357.84	0,03	105
all	6836.27	0.65	99	6866.77	0.05	218

0.00 means less than 0.001

difference (p-value of 0.004, less than 0.01) between SOTA and our proposal for $p = 25$ and 30, meaning that there is significant evidences to reject the hypothesis that the GRASP with EvRPR and the SOTA attain similar objective function values.

To sum it up, the experimentation shows that our proposal provides the best trade-off between solution quality and computational running times.

5.3. Case study

This section details the strategic expansion of a Spanish company aiming to enhance its distribution network across the national territory. As part of this expansion, the company seeks to identify the best locations to host new warehouses and to improve product distribution efficiency. Given the dynamic nature of customer locations and the need for a robust preliminary approach, we considered the p-dispersion problem as a first approximation for warehouses location. Additionally, considering that the company is already operating and has some warehouses set, we address the Conditional p-Dispersion Diversity Problem to account for existing warehouse locations in our optimization model.

Due to the confidentiality clause with the consulting company, OGA, for which we solved this problem, we cannot disclose the identity of the customer, but we can mention that it belongs to the food industry. This Spanish company has experienced significant growth and aims to expand its logistical footprint to better serve its customer base. The primary objective of this project is to determine new warehouse locations that maximize geographic diversity, ensuring broad coverage and efficient distribution capabilities. We therefore considered to apply a diversity optimization model based on maximize the inter-distance minimum value since it is well documented that this model allows for the identification of locations that are widely spread out, thus providing a diverse and strategically advantageous distribution network (Parreño et al., 2021; Lu et al., 2023).

Given that the company already operates several warehouses, we consider the conditional p-dispersion model. It must be noted that this model perfectly fits the typical expansion scenario in which some locations are already selected (due to the current operations of the company), and we have to select some new sites to maximize overall distribution diversity while incorporating these pre-established nodes.

Key steps in our methodology include:

- Data Collection. Gathering geographical and logistical data relevant to potential warehouse sites across the national territory.
- Problem Formulation. Defining the Conditional p-Dispersion Problem to incorporate existing warehouse locations into our model.
- Optimization. Applying metaheuristic techniques to identify new sites maximally dispersed. In particular, we ran the GRASP Evolutionay PR algorithm to solve the problem.
- Evaluation. Assessing the proposed locations for feasibility, accessibility, and alignment with the company's strategic goals.

The instance provided by OGA contained 800 potential sites for warehouses over the Spanish territory. From this set of sites, 150 are already selected and they are currently operating, providing service to their customers. In their expansion plan, the company considers to select 50 new sites to cover the entire Spanish peninsula. At this stage, the set of customers is not defined and so they resort to geographical information to identify scatter and well-distributed

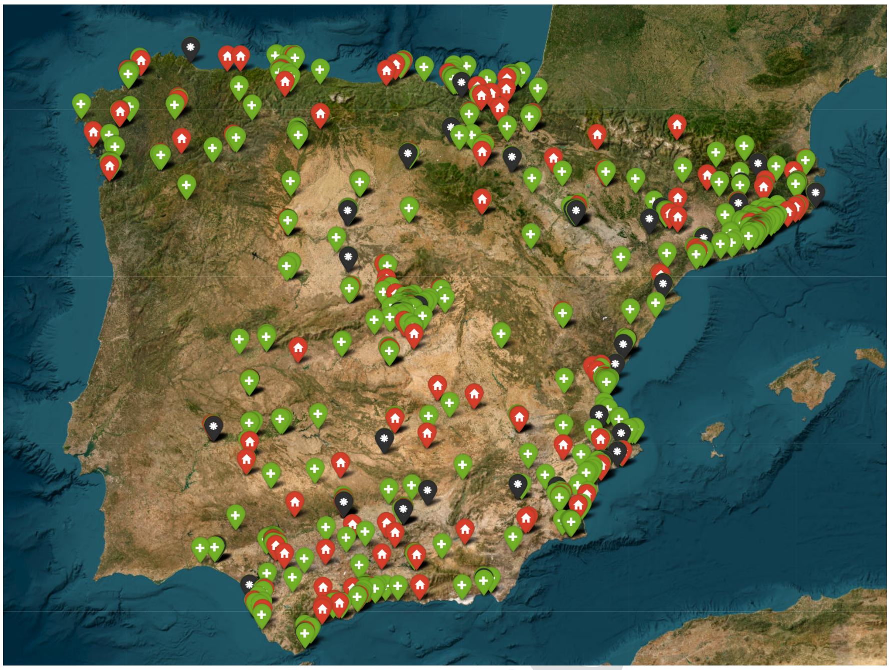

Figure 7: Candidate sites for warehouse location. Red bullet: existing warehouse locations (Q). Black bullet: new sites (P).

points, as identified by our model. Our heuristic method based on the evolutionary path relinking methodology is able to provide a high-quality solution in 1 minute of computing time. Figure 7 shows the solution on a map. By solving the Conditional p-Dispersion Problem, we provide a robust framework for identifying optimal warehouse locations, ensuring the company's continued growth and operational efficiency in the Spanish market.

6. Conclusion

This paper solves a problem known as the c-pDP recently proposed by Cherkesly and Contardo (2021). The c-pDP is a variant of the well-known dispersion problem that aims to select a set of p elements when q elements has been already selected while maximizing the dissimilarity among the elements.

This problem is $\mathcal{NP}$-hard, which makes exact methods impractical for large scale instances. In this research, we propose different metaheuristics, from the simplest one (GRASP) to a set of more complex metaheuristics focused on combining GRASP with PR, such as: static GRASP with PR, dynamic GRASP with PR and GRASP with EvPR. In the context of path relinking, instead of implementing the standard variant, we propose a more intelligent strategy that combines interior and exterior PR, that is, the RPR, to intensify the search (when two solutions are different) or diversity it (when two solutions are similar).

Many combinatorial optimization problems, and in particular the c-pD, that present flat landscapes where it is easy to find several different solutions presenting the same objective function value pose a challenge to heuristic methods. In order to overcome this difficulty, a new diversity metric has been proposed to evaluate the similarity of two solutions and in such a way, to allow the algorithm to escape from suboptimal basin of attractions. The metric has been coined as the δ-metric along this work and it is not only valid to solve the c-pD problem, but also to solve any combinatorial optimization problem in which the objective value is a max-min objective function (or a min-max objective function).

Computational results conclude that our proposal, GRASP with EvRPR, is able to solve large and complex instances in short computational time. A comparison with the state-of-the-art algorithm shows that both have a similar performance but our algorithm exhibits significantly in shorter times. Additionally, our experimentation shows the suitability of this method to solve a practical case from a Spanish company in the food industry. To conclude, GRASP with EvRPR is suitable to solve both academic and realistic problems.

Acknowledgement

This research has been partially supported by the Ministerio de Ciencia e Innovación of Spain (Grant Ref. PID2021-125709OB-C21 and PID2021-125709OA-C22) funded by MCIN /AEI /10.13039/501100011033 / FEDER, UE. It has been also supported by the Generalitat Valenciana (CIAICO/2021/224).

A.D. López-Sánchez acknowledges support from the Spanish Ministry of "Economía, Industria y Competitividad" through Project PID2019-104263RB-C41 and of "Ministerio de Ciencia e Innovación" through Project PID2022-139543OB-C41.

The authors wish to express their gratitude to Cherkesly and Contardo for their collaboration in the provision of the executable of their code to perform a fair comparison.

Furthermore, we would like to thank to OGA for generously sharing their data, and especially Arturo López-Damas for his technical help with the experiments in this project.

CRediT authorship contribution statement

Jesús Sánchez-Oro: Methodology, Algorithm design, Algorithm implementation. **Anna Martínez-Gavara:** Methodology, Algorithm design, Writing. **Ana D. López-Sánchez:** Methodology, Algorithm design, Writing. **Rafael Martí:** Methodology, Writing. **Abraham Duarte:** Methodology, Writing the revised paper.

References

Adil, G.K., Ghosh, J.B., 2005. Maximum diversity/similarity models with extension to part grouping. International Transactions in Operational Research 12, 311–323.

Cherkesly, M., Contardo, C., 2021. The conditional p-dispersion problem. Journal of Global Optimization 81, 23–83.

Contardo, C., 2020. Decremental clustering for the solution of p-dispersion problems to proven optimality. INFORMS Journal on Optimization 2, 134–144.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research 7, 1–30.

Erkut, E., 1990. The discrete p-dispersion problem. European Journal of Operational Research 46, 48–60.

Erkut, E., Neuman, S., 1989. Analytical models for locating undesirable facilities. European Journal of Operational Research 40, 275–291.

Feo, T.A., Resende, M.G.C., Smith, S.H., 1994. A greedy randomized adaptive search procedure for maximum independent set. Operations Research 42, 860–878.

Glover, F., 2014. Exterior path relinking for zero-one optimization. International Journal of Applied Metaheuristic Computing 5(3), 8.

Glover, F., Ching-Chung, K., Dhir, K.S., 1995. A discrete optimization model for preserving biological diversity. Applied mathematical modelling 19, 696–701.

Glover, F., Kuo, C.C., Dhir, K.S., 1998. Heuristic algorithms for the maximum diversity problem. Journal of information and Optimization Sciences 19, 109–132.

Glover, F., Laguna, M., 1997. Tabu search. Kluwer.

Kuby, M.J., 1987. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. Geographical Analysis 19, 315–329.

Kudela, J., 2020. Social distancing as p-dispersion problem. Ieee Access 8, 149402–149411.

Laguna, M., Martí, R., 1999. Grasp and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 11, 44–52.

Lozano-Osorio, I., Martínez-Gavara, A., Martí, R., Duarte, A., 2022. Max–min dispersion with capacity and cost for a practical location problem. Expert Systems with Applications 200, 116899.

Lozano-Osorio, I., Sánchez-Oro, J., López-Sánchez, A.D., Duarte, A., 2023. A reactive path relinking algorithm for solving the bi-objective p-median and p-dispersion problem. Soft Computing , 1–31.

Lu, Z., Martínez-Gavara, A., Hao, J.K., Lai, X., 2023. Solution-based tabu search for the capacitated dispersion problem. Expert Systems with Applications 223, 119856.

Martí, R., Martínez-Gavara, A., Pérez-Peló, S., Sánchez-Oro, J., 2022. A review on discrete diversity and dispersion maximization from an or perspective. European Journal of Operational Research 299, 795–813.

Parreño, F., Álvarez-Valdés, R., Martí, R., 2021. Measuring diversity. A review and an empirical analysis. European Journal of Operational Research 289, 515 – 532.

Rahman, M., Kuby, M., 1995. A multiobjective model for locating solid waste transfer facilities using an empirical opposition function. INFOR: Information Systems and Operational Research 33, 34–49.

Reinelt, G., 1991. TSPLIB — A Traveling Salesman Problem library. ORSA Journal on Computing 3, 376–384.

Resende, M., Martí, R., Gallego, M., Duarte, A., 2010. GRASP and path relinking for the max–min diversity problem. Computers & Operations Research 37, 498–508.

Resende, M.G.C., Werneck, R.F., 2004. A hybrid heuristic for the p-median problem. Journal of Heuristics 10, 59–88.

Weitz, R., Lakshminarayanan, S., 1997. An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups, vlsi design, and exam scheduling. Omega 25, 473–482.

Weitz, R.R., Lakshminarayanan, S., 1998. An empirical comparison of heuristic methods for creating maximally diverse groups. Journal of the operational Research Society 49, 635–646.