Available online at www.sciencedirect.com

ScienceDirect

Information Sciences 00 (2019) 1–21

Information Sciences

Seeking Affinity Structure: Strategies for Improving m-best Graph Matching

Manuel Curado[a,*], Francisco Escolano[b], Miguel A. Lozano[b], Edwin R. Hancock[c]

[a] *Department of Technology, Catholic University of Avila, 05005 Avila, Spain*
[b] *Department of Computer Science and AI, University of Alicante, 03690 Alicante, Spain*
[c] *Department of Computer Science, University of York, UK*

Abstract

State-of-the-art methods for finding the m-best solutions to graph matching (QAP) rely on exclusion strategies. The k-th best solution is found by excluding all better ones from the search space. This provides diversity, a natural requirement for transforming a MAP problem into a m-Best one. Since diversity enforces mode hopping, it is usually combined with a mode-approximation strategy such as marginalisation. However, these methods are generic insofar they do not incorporate the detailed structure of the problem at hand, i.e. the properties of the global affinity matrix which characterise the search space. Without this knowledge, it is thus hard to devise a practical criterion for choosing the next variable to clamp. In this paper, we propose several strategies to select the next variable to clamp, spanning the whole range between depth-first and breadth-first search, and we contribute with a unifying view for characterising the search space on the fly. Our strategies are: a) number of factors in which the variables participate, b) centrality measures associated with the affinity matrix, and c) discrete pooling. Our experiments show that max number of factors and centrality provide a trade-off between efficiency and accuracy, whereas discrete pooling leads to an improvement of the state-of-the-art.

Keywords: m-Best Graph Matching, Binary-Tree Partitions, QAP

1. Introduction

1.1. Motivation

The importance of finding the m-best solutions to discrete optimization problems has been appreciated for four decades [15][20][29][8][1][13][12]. However, its practical utility for solving computer vision problems has not been demonstrated until recently [21][2][26][24][23]. The principle underpinning this utility is the fact that the MAP (Maximum a Posteriori) solution provides a single low-energy configuration. Such a configuration might be very far from the ground-truth, as it has been shown in large-scale empirical studies concerning stereo vision [19] and other low-level vision problems [27]. A promising alternative is to capture the multi-modal structure of the energy landscape, for instance through enforcing *diverse solutions*. As an example, in [2], m qualitatively different highly probable solutions are obtained through mode hopping. This is done by formulating the m-best mode problem in terms of providing solutions that are at least k units dissimilar from the MAP solution. If one uses a dot-product dissimilarity function such as the Hamming distance, m-mode selection reduces to the original MAP problem whose unary potentials (e.g. data terms) are modified (perturbed) while the binary potentials (e.g. the edge or discontinuity label process) are left

[*] Corresponding author: Tel.: +34 653164473;
Email address: manuel.curado@ucavila.es (Manuel Curado)

unchanged. This provides solutions that are biased away from the current solution.

The m-best mode algorithm and its variations [2][13] are *direct* methods because they implicitly enforce diversity by penalizing their original energy functions. Such a design does not necessarily impose any sequentiality (for instance to obtain the 3^{rd} best solution after having the 1^{st} and the 2^{nd} ones). A recent formulation of the *joint m-best diverse problem* [12] provides a nice set of nested solutions and efficient algorithms as well. However, it is limited to binary labellings.

On the other hand, *indirect* methods drive the location of the k-th best solution through iterative exclusion with respect to the prior $k-1$ solutions. While these methods are forced to be sequential, some of them provide a characterization of the search space for the k-th solution after excluding the prior $k-1$ solutions. For instance, the well-known STRIPES method [8] derives a condition that defines the marginal polytope resulting from excluding the 1^{st} best solution. This condition produces an exact characterization of the so-called assignment-excluded maximal polytope when the factors defining the MRF define a tree. Consequently, in the general case, i.e. when the MRF defines a forest it is applied to a set of spanning trees. This is the STRI (Spanning Tree Inequality) part of the algorithm. The PES (Partitioning for Enumerating Solutions) part consists of partitioning the search space using constraint satisfaction as in [15][20][29]. Given any variable whose value is different in the two previous solutions, the method selects one of them randomly and explores two possibilities while preserving the previous constraints, namely a) having a given value and b) having a different value.

Regarding the Quadratic Assignment Problem (QAP) or graph matching, there have been few efforts for providing an m-best approach. The most significant one is that of Rezatofighi et al. [23] which drives on [24] where a Partition Enumeration Strategy (PES) is combined with an efficient exclusion strategy. Although the PES is driven by the selection of a particular matching among the ones differing from previous solutions, such a selection is done randomly for the sake of compensating the depth-first behaviour of the Binary Tree Partitioning construction which supports the m-best search.

For that, these methods for finding the m-best solutions to graph matching (QAP) rely on exclusion strategies, where the k-th best solution is found by excluding all better ones from the search space, providing diversity and combining with a mode-approximation strategy such as marginalisation. However, these methods are generic insofar they do not incorporate the detailed structure of the problem, as the properties of the global affinity matrix which characterise the search space. We propose to study different criteria for selecting the next variable to clamp.

1.2. Contributions

In this paper, we fill this gap of the state of the art by proposing several strategies for variable selection. We study how works the state-of-the-art methods for finding the m-best solutions to graph matching rely on exclusion strategies, where different solutions are found by excluding all better ones from the search space. These methods are generic insofar they do not incorporate the detailed structure that characterise the search space. Our working hypothesis is that the structure of the problem, which is encoded by the affinity matrix, provides valuable information for focusing on informative subspaces. For that, it is thus hard to devise a practical criterion for choosing the next variable to clamp (the state-of-the-art methods use random criterion)

The main novelty of our paper is to propose four strategies to select the next variable to clamp. Our main goal is to obtain better results with our proposed strategies that improve over the state-of-the-art criterion (random strategy) with a smaller number of solutions, and reducing the running time. In summary, we: i) span the whole range of possibilities between depth-first and breadth-first search, ii) we obtain a unifying view for characterising the search space, and iii) we improve over the state-of- the-art methods in terms of both efficiency and accuracy.

Another objective is to establish a discussion about the importance of devising a correct strategy of variable selection, opening the door to study the different proposal of variable selection in m-best approaches which that incorporate a better balanced search between depth and breath, reaching the optimal solution in fewer iterations.

For that, in this paper, in Section 2, we will formulate the problem of m-Best Graph Matching and we will present our main contributions in Section 3: four variable selection strategies to characterise the search space of m-best problems (*max factor, discrete pooling, median discrete pooling and trimmed centrality*). In Section 4 is devoted to an in-depth experimental analysis of a challenging standard benchmark. Our experiments confirm the hypothesis and some of the strategies improve the state-of-the-art both in quantitative and efficiency terms, wheres others are competitive with the state-of-the-art in terms of accuracy but provide better running-times. For that, we have studied: i) the comparison between of all our strategies with a several methods of the state of the art, showing how our strategies improve the remaining methods in accuracy and running time, ii) the behaviour of the m-best approach for a large number of solutions, obtaining that our strategies tolerate a high number of outliers, characterizing the search space (providing a trade-off between breadth-first and depth-first search and recovering earlier from a wrong variable selection), iii) the efficiency of our methods in terms of how many solutions are needed to outperform the state of the art (modes and running time), and iv) the analysis of all pairs by spectral characterization, explaining how our best strategy finds an intelligent trade-off between depth-first and breadth-first search. Finally, in Section 5 we summarise our conclusions and future work.

2. *m*-Best Graph Matching

2.1. Graph Matching

Given two attributed graphs $G = (V, E)$ and $G' = (V', E')$, where V represents the set of nodes and E the set of edges, each node $v_i \in V$ or edge $e_{ij} \in E$ has an associated feature vector $\mathbf{f_i}$ or $\mathbf{f_{ij}}$. Let $n = |V|$ and $n' = |V'|$ the number of features in both graphs. For each pair of unary features $\mathbf{f_i}$ and $\mathbf{f'_k}$, associated respectively with nodes $v_i \in V$ and $v'_k \in V'$, there is a node affinity measure $\phi(\mathbf{f_i}, \mathbf{f'_k})$. Similarly, for each pair of edges $e_{ij} \in E$ and $e'_{kl} \in E'$ there is a pairwise affinity measure $\psi(e_{ij}, e'_{kl})$. The affinity structure is represented by a symmetric affinity matrix $K \in \mathbb{R}^{nn' \times nn'}$, where the diagonal elements $K_{iikk} = \phi(\mathbf{f_i}, \mathbf{f'_k})$ contain the unary affinities, whereas the off-diagonal elements $K_{ikjl} = \psi(e_{ij}, e'_{kl})$ contain the pairwise affinities.
Given the two graphs and the affinity matrix, the Graph Matching (GM) problem consists of finding the optimal binary assignment matrix $X \in \{0, 1\}^{n \times n'}$, where $X_{ik} = 1$ implies that node $v_i \in V$ corresponds to node $v'_k \in V'$, and $X_{ik} = 0$ otherwise. Herein, optimality refers to maximizing the quadratic function:

$$J(X) = \sum_{ik} X_{ik} \phi(\mathbf{f_i}, \mathbf{f'_k}) + \sum_{ikjl} X_{ik} X_{jl} \psi(e_{ij}, e'_{kl}) \ , \tag{1}$$

which naturally leads to the Lawler's formulation of the Quadratic Assignment Problem (QAP) [14]:

$$\begin{aligned} J(X) &= \sum_{ik} X_{ik} K_{iikk} + \sum_{ikjl} X_{ik} X_{jl} K_{ikjl} \\ &= \mathbf{x}^T K \mathbf{x} \equiv J(\mathbf{x}) \ , \end{aligned} \tag{2}$$

where $\mathbf{x} = \mathrm{vec}(X) \in \mathbb{R}^{nn'}$ is a vectorization of X given by the concatenation of its columns. In addition, since the solution X is constrained to be a one-to-one mapping, we have the following integer quadratic program (IQP):

$$\begin{aligned} \mathbf{x}^* &= \arg\max_{\mathbf{x}} (\mathbf{x}^T K \mathbf{x}) \\ s.t. & \quad \begin{cases} \mathbf{x} \in \{0, 1\}^{n \times n'} \\ \forall i \ \sum_{k=1}^{n'} \mathbf{x}_{ik} \leq 1 \ , \quad \forall k \ \sum_{i=1}^{n} \mathbf{x}_{ik} \leq 1 \ , \end{cases} \end{aligned} \tag{3}$$

which is known to be an NP-hard problem. As a result, its tractability relies on convenient relaxations. The most common relaxation is to transform the IQP into a non-convex quadratic problem by both making $\mathbf{x} \in [0, 1]^{n \times n'}$ (i.e. continuous) and transforming the two-way constraints into doubly-stochastic matrices (the sum of both their rows and columns must be the unit) [9][30][28][33]. As an alternative, dropping doubly-stochasticity and imposing the unit norm $||\mathbf{x}||^2 = 1$ leads to spectral relaxations, since $\mathbf{x}^T K \mathbf{x}$ can be interpreted as the Rayleigh quotient [16][5][3]. The third group of approaches rely on convex relaxations. For instance, in [25] a new variable $Y = \mathbf{x}\mathbf{x}^T$ is constrained to be semi-definite which leads to a convex program, whereas in [28] the objective function is transformed into a convex function.

2.2. *m*-Best Graph Matching

Relaxing the IQP may lead us to miss the global optimum $\mathbf{x}^*$. In [18], Lyzinski et al. prove that this is the case for convex relaxations, especially when the graphs being matched are uncorrelated[1]. For non-convex relaxations, we have that $J(\mathbf{x}) = \mathbf{x}^T K\mathbf{x}$ is thus multi-modal, which in turn compromises the quality of the solutions obtained by gradient ascent methods.

Enforcing Diversity. In a multi-modal scenario, the maximum a posteriori (MAP) criterion is forced to choose a solution $\mathbf{x}^1 = \arg\max_{\mathbf{x}}(\mathbf{x}^T K\mathbf{x})$ where $\mathbf{x}^1 \neq \mathbf{x}^*$, in general. It is then more convenient to retain a set of $m > 1$ modes. Finding the m-Best solution to the QAP problem can be enforced by adding *diversity constraints* to the formulation in Eq. 3. These constraints take the following shape:

$$\Delta(\mathbf{x}, \mathbf{x}^{m'}) \geq k_{m'} \;\; \forall m' \in \{1, \ldots, m-1\} \;, \tag{4}$$

where $\mathbf{x}^{m'}$ is the m'-best solution, $\Delta(\mathbf{x}, \mathbf{x}^{m'})$ is a dissimilarity function, and $k_{m'}$ establishes a minimum degree of similarity with respect to previously chosen solutions [2]. For instance, let $\Delta(\mathbf{x}, \mathbf{x}^{m'}) = \sum_a [\![\mathbf{x}_a \neq \mathbf{x}_a^{m'}]\!]$, with $a = ik$, the Hamming distance ($[\![.]\!]$ is thus an indicator function). Then, from [22] and [33] we obtain

$$\begin{aligned} \mathbf{x}^m &= \arg\max_{\mathbf{x}} J_\Delta(\mathbf{x}) \\ &= \arg\max_{\mathbf{x}} \sum_a \left(K_a + \sum_{m'=1}^{m-1} \lambda_{m'} [\![\mathbf{x}_a \neq \mathbf{x}_a^{m'}]\!] \right) \mathbf{x}_a \\ &\quad + \mathrm{tr}(K_q^T Y) \;, \end{aligned} \tag{5}$$

where $\lambda_{m'}$ are Lagrange multipliers, $K_a = K_{iikk}$, K_q is the off-diagonal of K, and $Y \in \{0,1\}^{nn' \times nn'}$ is an edge compatibility matrix. Therefore, diversity can be enforced through the deterministic perturbation of the unary affinities. As a consequence, $J(\mathbf{x})$ becomes $J_\Delta(\mathbf{x})$.

Marginalization. The second ingredient of m-Best QAP aims to cope with the uncertainty of the prediction, and thus it is probabilistic. In certain situations, some correspondences $\mathbf{x}_a$ are either disclosed through learning [10] or predicted by other algorithms (such as convex relaxations). With these correspondences at hand, the joint probability $P(\mathbf{x}) = \frac{1}{Z} e^{J(\mathbf{x})}$ with partition function Z, becomes more tractable. As an alternative, the tractability of the joint probability can be improved by computing the marginals of a subset of correspondences. In practice, where Z is not available, the marginalization captures some unnormalised slices of the joint probability. In this regard, marginals, such as $\mathrm{marg}(\mathbf{x}_a) = \sum_{\mathbf{x}, \mathbf{x}_a = 1} \exp(J(\mathbf{x}))$, or max-marginals $\max - \mathrm{marg}(\mathbf{x}_a) = \max_{\mathbf{x}, \mathbf{x}_a = 1} \exp(J(\mathbf{x}))$ are often intractable in Graph Matching. Since for $m > 1$ we pool a set of modes $\mathbf{x}^1, \ldots, \mathbf{x}^m$, we define the sets $\mathcal{X}_a = \{\mathbf{x}^{m'} : \mathbf{x}_a^{m'} = 1 \;\; \forall m' \in \{1, \ldots, m\}\}$. Then, the so-called *mode marginals* express the whole spectrum between marginals and max-marginals [22]. Mode marginals are defined as the probabilistic support of each particular matching x_a:

$$\Phi(x_a) = \sum_{\mathbf{x}^{m'} \in \mathcal{X}_a} \frac{\exp(J_\Delta(\mathbf{x}^{m'}))}{\sum_{\mathbf{x}^t \in \mathcal{X}_a} \exp(J_\Delta(\mathbf{x}^t))} \;. \tag{6}$$

When $m = 1$ we have max-marginals, and when $\mathcal{X}$ is the whole matching space, we have marginals. Therefore, mode marginals provide a balanced strategy for fusing evidence coming from diverse modes. This strategy can be either applied online, i.e. at each mode seeking step, or off-line, i.e. as a bias of the whole process. The former choice is convenient when one wants to focus on a particular set of matchings, for instance those which are close enough to the disclosed ones, (see focused inference examples in [22]). The second choice (off-line) supports matching binarisation (clean-up) [23].

[1] To the best of our knowledge this is the first work that explicitly relates the properties of the graphs with the chance of reaching optimal solutions. We will come back to this point later on.

2.3. Sequential vs Joint m-Best

The interplay between diversification and marginalization leads to a systematic analysis of the search space in the neighbourhood of the MAP solution. In m-Best graph matching, where the marginals are usually computed offline, the critical element is diversity. To commence, it is hard to learn the values of $k_{m'}$ in Eq. 4 unless the topology of the energy landscape is unveiled (for instance through the spectral analysis of K). As a result, the $\lambda_{m'}$ multipliers cannot provide a good trade-off between energy and diversity. Secondly, the incremental (greedy) update of $\mathbf{x}^m$ does not guarantee that all the m modes are placed in local maxima [11]. Recent joint m-Best methods [12], which do not enforce the MAP solution as part of the set of solutions, outperform sequential ones both in terms of quality and runtime, but the difference in runtime grows with m. Thus, a trade-off between sequential and joint m-best can be implemented by expanding a tree-like structure, such as a binary tree partitioning (BTP) [24][23], but driven by *heuristic variable selection*. Herein we present variable selection strategies specifically tailored to the graph matching problem.

2.4. Illustrating the Alternatives

In a BTP Model, the binary tree partition algorithm is a PES (Partitioning for Enumerating Solutions) [8] strategy (see Fig. 1). It assumes that $\mathbf{x}^1$ and $\mathbf{x}^2$ are already known, they are different $\mathbf{x}^1 \neq \mathbf{x}^2$ and such a difference is at least one bit. Then, $\Delta(\mathbf{x}^1, \mathbf{x}^2) \equiv [(\mathbf{x}^1)^T\mathbf{x}^2 < ||\mathbf{x}^2||_1] > 1$. Since BTP starts by locating $\mathbf{x}^3$, let $\mathcal{X}^3$ be the void root of the tree: $\mathcal{X}^3$ spans two children $\mathcal{X}^{3,1}$ and $\mathcal{X}^{3,2}$ so that the former one must contain a mode satisfying $\Delta(\mathbf{x}^{3,1}, \mathbf{x}^1) > 1$ and the second child must contain a mode with $\Delta(\mathbf{x}^{3,2}, \mathbf{x}^2) > 1$. Then, these modes are obtained by triggering graph matching with a conveniently perturbed cost functions $J_\Delta(\mathbf{x})$, and the obtained $J_\Delta(\mathbf{x}^{3,1})$ and $J_\Delta(\mathbf{x}^{3,2})$ are respectively assigned to $\mathcal{X}^{3,1}$ and $\mathcal{X}^{3,2}$. For the next level (depth $d = 2$) there are two legal operations, namely a) *expand* two children from $\mathcal{X}^{3,*} = \arg\max_r\{J_\Delta(\mathbf{x}^{3,r})\}$, and b) *copy* the non-winning node. Therefore, for depth d, the BTP retains $d + 1$ *active* solutions and $\mathcal{X}^{d+2,*} = \arg\max_r\{J_\Delta(\mathbf{x}^{d+2,r})\}$.

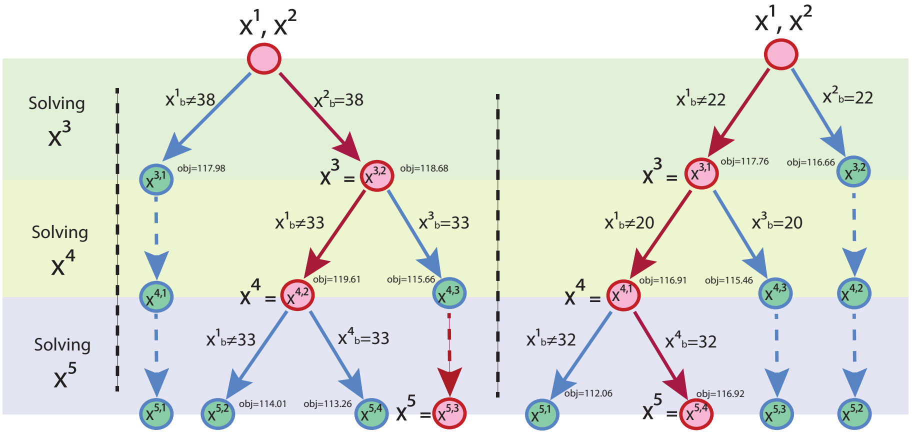


Figure 1. BTP example. In this figure, we show how works the selection of the following solution. In this example, we compare the BP random variable selection strategy (left) and another strategy different of random selection (right). Regarding the graphical notation, we differentiate between an expanded action in the tree (solid arrow) and a copy action (dotted arrow), and we show (in red) the path of m-best selected solutions, where a continued path means a probable promising subspace, whereas a discontinue one means a jump to another subspace (diversity). In each iteration (green, yellow and purple areas), we select a variable b through a heuristic (we propose four strategies in the following sections), and we expand two partial solutions from previous solutions and the best solution of the previous ones with respect to different constraints on b. Finally, we obtain an objective value of each partial solution and select the m-best solutions between all non-selected partial solutions.

The BTP model: Heuristic Variable Selection. BTP exploits the fact that $\Delta(\mathbf{x}^r, \mathbf{x}^s) > 1$ to focus on one of the different bits b. This bit, chosen at random, leads to complementary assignments. For instance, if $\mathbf{x}^1$ and $\mathbf{x}^2$ differ in b, then a convenient assignment to $\mathbf{x}^{3,1}$ before triggering graph matching is $\mathbf{x}^{3,1}_b = \mathbf{x}^1_b$. As a consequence, setting $\mathbf{x}^{3,2}_b = \mathbf{x}^2_b$ leads to: a) *complementary subspaces* associated with $\mathcal{X}^{3,1}$ and $\mathcal{X}^{3,2}$, and b) *redundant constraints*, since

Algorithm 1 Binary Tree Partition Algorithm

```
Input: $m$
Output: $\mathbf{x}^k, k = 1, ..., m$;
STEP 1. Find the $1^{st}$ best solution:
$\mathbf{x}^1 = \arg\max_{x \in \mathcal{X}} J_\Delta(x)$
$\mathcal{X}^1 = \mathcal{X}$ and $J_\Delta(\mathbf{x}^1)$ is assigned to $\mathcal{X}^1$
STEP 2. Find the $2^{nd}$ best solution:
$\mathcal{X}^2 = \{x \in \mathcal{X}^1 \mid \Delta(\mathbf{x^1}, \mathbf{x^2}) > 1\}$
$\mathbf{x}^2 = \arg\max_{x \in \mathcal{X}^2} J_\Delta(\mathbf{x})$
$\mathcal{X} = \mathcal{X}^1 \cap \mathcal{X}^2$ and $J_\Delta(\mathbf{x}^2)$ is assigned to $\mathcal{X}^2$
STEP 3a. Finding the $3^{rd}$ best solution: Select variable $b$ through a strategy, where the previous two first solutions differ in $\mathbf{x}_b^1 \neq \mathbf{x}_b^2$
STEP 3b. Split the feasible set into two disjoints sets as follows (expand action):
$\mathcal{X}^{3,1} = \{x \in \mathcal{X}^1 \mid \mathbf{x_b} \neq \mathbf{x_b^2}, \Delta(x, \mathbf{x}^1) > 1, \Delta(x, \mathbf{x}^2) > 1\}$
$\mathcal{X}^{3,2} = \{x \in \mathcal{X}^2 \mid \mathbf{x_b} = \mathbf{x_b^2}, \Delta(x, \mathbf{x}^2) > 1, \Delta(x, \mathbf{x}^1) > 1\}$
Remove redundant constraints (in orange)
STEP 3c. Select solution as follows:
$\mathbf{x}^{3,1} = \arg\max_{x \in \mathcal{X}^{3,1}} J_\Delta(\mathbf{x})$
$\mathcal{X}^1 = \mathcal{X}^{3,1} \cap \mathcal{X}^1$ and $J_\Delta(\mathbf{x}^{3,1})$ is assigned to $\mathcal{X}^{3,1}$
$\mathbf{x}^{3,2} = \arg\max_{x \in \mathcal{X}^{3,2}} J_\Delta(\mathbf{x})$
$\mathcal{X}^2 = \mathcal{X}^{3,2} \cap \mathcal{X}^2$ and $J_\Delta(\mathbf{x}^{3,2})$ is assigned to $\mathcal{X}^{3,2}$
Assign $\mathbf{x}^3 = \arg\max_r\{J_\Delta(\mathbf{x}^{3,r})\}$
Insert the non-assigned solution to $\mathbf{x}^3$, with value $J_\Delta(\mathbf{x}^{3,*})$, in a vector $y$ (copy action)
repeat
    STEP 4a. Finding the $k$-th best solution: Select b (as in Step 3a) where the previous solution $k-1$ and the best solution between the selected solutions $l$ ($l \neq k-1$) differ in $\mathbf{x}_b^l \neq \mathbf{x}_b^{k-1}$. Then, expand two disjoints sets $\mathcal{X}^{k,1}$ and $\mathcal{X}^{k,2}$ as in Step 3b with the following constraints (remove orange ones):
    $\mathcal{X}^{k,r} = \{x \in \mathcal{X}^l \mid \mathbf{x_b} \neq \mathbf{x_b^{k-1}}, \Delta(x, \mathbf{x}^l) > 1, \Delta(x, \mathbf{x}^{k-1}) > 1\}$
    $\mathcal{X}^{k,k-1} = \{x \in \mathcal{X}^{k-1} \mid \mathbf{x_b} = \mathbf{x_b^{k-1}}, \Delta(x, \mathbf{x}^{k-1}) > 1, \Delta(x, \mathbf{x}^l) > 1\}$
    STEP 4b. Select $\mathbf{x}^{k,r}$ and $\mathbf{x}^{k,k-1}$ from $\mathcal{X}^{k,r}$ and $\mathcal{X}^{k,k-1}$, respectively (as in Step 3c), and assign $\mathbf{x}^k = \arg\max\{J_\Delta(\mathbf{x}^{k,1}), J_\Delta(\mathbf{x}^{k,2}), y\}$. If $\mathbf{x}^k \in y$, remove it from $y$.
    Insert the non-assigned solution to $\mathbf{x}^k$, with value $J_\Delta(\mathbf{x}^{k,*})$, in vector $y$ (copy action)
until $k > m$
```

$\mathbf{x}_b^{3,1} = \mathbf{x}_b^1$ leads implicitly to $\Delta(\mathbf{x}^{3,2}, \mathbf{x}^2) > 1$ and the constraint $[(\mathbf{x}^{3,1})^T\mathbf{x}^2 < ||\mathbf{x}^2||_1] > 1$ is no longer necessary; symmetrically the assignment $\mathbf{x}_b^{3,2} = \mathbf{x}_b^2$ makes $[(\mathbf{x}^{3,2})^T\mathbf{x}^1 < ||\mathbf{x}^1||_1] > 1$ redundant.

The BTP model (see Algorithm 1) provides a) complementary subspaces, which yields a balanced breadth-wise organisation as depth increases, and b) redundancy, which ensures the tractability of the m-best graph matching for large values of m. However, the random selection of differing bits is not criticised in [24][23]. It basically follows up the approaches in [15][20][29]. However, the main contribution of this approach is to show that heuristic variable selection (in red in Alg. 1) provides a significant room for improvement (both in quantitatively and running-time terms).

2.4.1. Variable Selection

The BTP model. BTP exploits the fact that $\Delta(\mathbf{x}^r, \mathbf{x}^s) > 1$ to focus on one of the different bits b. This bit, chosen at random, leads to complementary assignments. For instance, if $\mathbf{x}^1$ and $\mathbf{x}^2$ differ in b, then a convenient assignment to $\mathbf{x}^{3,1}$ before triggering graph matching is $\mathbf{x}_b^{3,1} = \mathbf{x}_b^1$. As a consequence, setting $\mathbf{x}_b^{3,2} = \mathbf{x}_b^2$ leads to: a) *complementary subspaces* associated with $\mathcal{X}^{3,1}$ and $\mathcal{X}^{3,2}$, and b) *redundant constraints*, since $\mathbf{x}_b^{3,1} = \mathbf{x}_b^1$ leads implicitly to

$\Delta(\mathbf{x}^{3,2}, \mathbf{x}^2) > 1$ and the constraint $[(\mathbf{x}^{3,1})^T\mathbf{x}^2 < ||\mathbf{x}^2||_1] > 1$ is no longer necessary; symmetrically the assignment $\mathbf{x}_b^{3,2} = \mathbf{x}_b^2$ makes $[(\mathbf{x}^{3,2})^T\mathbf{x}^1 < ||\mathbf{x}^1||_1] > 1$ redundant.

The BTP model provides a) complementary subspaces, which yields a balanced breadth-wise organisation as depth increases, and b) redundancy, which ensures the tractability of the m-Best graph matching for large values of m. However, the random selection of differing bits is not criticised in [24][23]. It basically follows up the approaches in [15][20][29]. However, *the main contribution of this paper is to show that heuristic variable selection provides a significant room for improvement (both in quantitatively and running-time terms).*

3. Proposing Alternative Strategies for Variable Selection

In Fig. 2, we show a toy example of graph matching in order to illustrate the alternative strategies for variable selection. Given the 1st (b) and 2nd (m) best solutions, which differ in the variables v_2 and v_3 (bits $2b$ and $3d$ in $\mathbf{x}^m$, and bits $2d$ and $3b$ in $\mathbf{x}^b$), the Binary Tree Partition (BTP) must create a root node and expand two candidates for the 3rd best solution: one of the coming from branch b and the other one from branch m. If we select the variable v_2 as the most promising one, then the branch b will explore new solutions with $v_2 \neq b$, whereas the branch m will focus on new solutions with $v_2 = b$. Similarly, if the variable v_3 is selected, the branch b will explore new solutions with $v_3 \neq d$, and the branch m will focus on new solutions with $v_3 = d$.

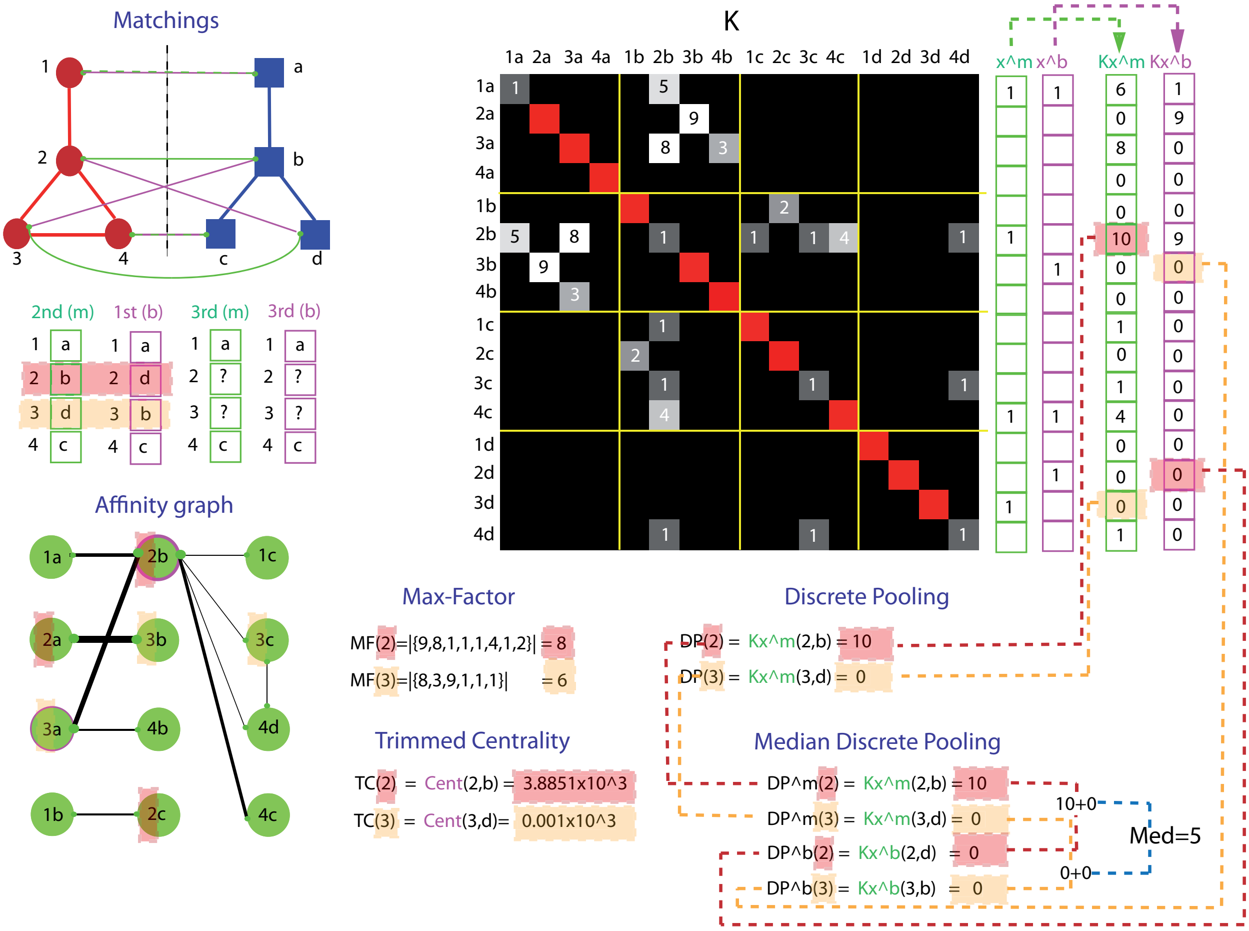


Figure 2. Toy example and graphical summary. Top left: graph matching example with 1st and 2nd best solutions. Top right: detailed structure of the affinity matrix K as well as its interactions with $\mathbf{x}^m$ and $\mathbf{x}^b$. These interactions lead to the strategies DP and MDP. Bottom right: edges and nodes of the affinity matrix, leading to the strategies MF and TC. Bottom right: numerical results for each strategy (see text).

3.1. Max Factor

The first proposed strategy (MF) is based on the MRF notation, $\theta_i(y_i) = K_{ii,y_iy_i}$ and $\theta_{ij}(y_i, y_j) = K_{iy_i,jy_j} + K_{jy_j,iy_i}$ specify respectively the unary and pairwise potentials associated with variables i and j. If two variables i and j are related through a pairwise potential $\theta_{ij}(y_i, y_j)$, then some of the entries K_{iy_i,jy_j} should be non-zero. These entries represent edge compatibilities between ij and y_i, y_j. We can thus define how influential is the variable i associated with the node $v_i \in \mathcal{V}$ in terms of how many potential matchings are available in K:

$$\mathbf{I}_{MF}(i) = \sum_{j=1}^{|\mathcal{V}|} [\![K_{iy_i,jy_j} \neq 0]\!]_{y_i \neq i, y_j \neq j} \, , \tag{7}$$

that is, $\mathbf{I}_{MF}(i)$ is not exactly the degree $deg(v_i)$ of $v_i \in V$ but the number of potential rectangles (i, y_i, j, y_j) rooted on the node v_i. This is consistent with the Graduated Assignment method [9] which maximizes the number of these rectangles. Therefore, let $\mathcal{D}(\mathbf{x}^m, \mathbf{x}^b)$ be the set of indices $a = ik$ associated with the bits where two candidate solutions $\mathbf{x}^m$ and $\mathbf{x}^b$ differ. Then, the *max-factor* (MF) strategy will select the variable $v = \arg\max_{ik \in \mathcal{D}(\mathbf{x}^m, \mathbf{x}^b)} \mathbf{I}_{MF}(i)$, where $v \in \mathcal{V}$, i.e. it represents a node and i is the index of a vector $\mathbf{v}$ where $\mathbf{v}_i = k$. Therefore, herein we prefer to represent the matchings ik as $|\mathcal{V}|$-dimensional vectors $\mathbf{v}$.

In Fig. 2, given the *affinity graph* (Fig. 2-Bottom left) where candidate matchings are the nodes and their *candidate rectangles* are the edges[2], we have that the variable v_2 is involved in nodes $2a$, $2b$ and $2c$. Since the node $2b$ is a hub in the affinity graph, it can potentially close 6 rectangles (links in affinity matrix): $(2,b,1,a)$, $(2,b,3,a)$, $(2,b,1,c)$, $(2,b,3,c)$, $(2,b,4,c)$, and $(2,b,4,d)$. In addition, the nodes $2a$ and $2c$ contribute with one rectangle: $(2,a,3,b)$ and $(2,c,1,b)$ respectively. Then there are 8 factors associated with v_2 . Similarly, we have 5 factors associated with v_3 , which is involved in nodes $3a$, $3b$ and $3c$. Consequently, the leading variable for MF is v_2 .

3.2. Trimmed Centrality

Our second proposed strategy for selecting the next variable to clamp is inspired by spectral methods. It is well known that these approaches to graph matching [16][5][3] rely on the principal eigenvector $\mathbf{v}_1$ of K, i.e. the one maximising the Rayleigh quotient: $\Psi_1 = \arg\max_{\Psi} \Psi^T K \Psi$. This is the Perron-Frobenius vector and its components can be interpreted as cluster memberships which tend to establish agreement links [16]. However, according to the spectral perturbation theory, only small perturbations of K are tolerated, and structured perturbations cause wrong assignments to belong to strong clusters. Only the Reweighted Random Walks Matching [3] (RRWM), where absorbing nodes soak affinity, tolerate a larger amount of noise. However, their performance is still far from the state-of-the-art.

Following this line of work we propose to increment the number of variational modes (eigenvectors) so that higher-order correlations in K are considered. As an alternative to RRWM, where random walks are dominated by degree statistics, centrality measures are more general and robust. For instance, in [6] subgraph centrality accounts for the participation of each node in all the subgraphs in a graph. When this concept is applied to the affinity matrix K we have that the relevance of a matching $a = ik$ is

$$\mathbf{I}_{TC}(a) = \sum_{j=1}^{p} \Psi_j(a) \exp(\lambda_j) \; , \tag{8}$$

where $K\Psi_j = \lambda_j \Psi_j$ and, in principle, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{nn'}$. Since solving the eigensystem takes $O((nn')^3)$, we will retain only the $p \ll n$ largest eigenpairs. In principle, this approximation is acceptable since K is indefinite in general. This implies that many λ_j's are negative and it contributes to a significant decay of $\exp(\lambda_j)$.
With the relevances $\mathbf{I}_{TC}(a)$ for all matchings $a = ik$ at hand, let $\mathcal{D}(\mathbf{x}^m, \mathbf{x}^b)$ be the set of indices defined above (bits where $\mathbf{x}^m$ and $\mathbf{x}^b$ differ). Then, the *trimmed centrality* (TC) will select $v = \arg\max_{ik \in \mathcal{D}(\mathbf{x}^m, \mathbf{x}^b)} (\max_k \mathbf{I}_{MF}(ik))$: for each matching ik in which i is involved we retain the k yielding the maximum TC for ik, and finally we select the variable associated with the maximum of all these maxima.

The *affinity graph* is also useful for understanding TC. Subgraph centrality accounts for the participation of each node in all the possible subgraphs. As we will detail below, the participation of a node is measured by the number of closed walks of all lengths that start and end at this node. Since the affinity graphs come from the matrix K whose diagonal is not zero in general, we consider than non-zero elements in this diagonal lead to self-loops in the affinity graph. In this regard, the most central node is $2b$ (3.8851×10^3) followed by the node $3a$ (2.2677×10^4) because of the strength of its tie with node $2a$ (both nodes encircled in magenta).
For the solution m in Fig. 2-Top left, we have that $v_2 = b$. Then its assigned TC is that of edge $2b$ (3.8851×10^3). However since whereas $v_3 = d$ for the solution m, we consider the TC of $3d$, which is the unit since $3d$ is an isolated node in the affinity graph and it only participated in self-loops. Therefore, TC selects v_2 . This result is consistent with the relative position of $2b$ in the affinity graph.

3.3. Discrete Pooling

The following proposed selection strategy is inspired in the gradient of the cost function $J(\mathbf{x})$. The first-order Taylor expansion around a given solution $\mathbf{x}^m$ leads to

$$J(\mathbf{x}) = \mathbf{x}^T K \mathbf{x} \approx J(\mathbf{x}^m) + (\mathbf{x} - \mathbf{x}^m) K \mathbf{x}^m \; , \tag{9}$$

[2] Nodes, i.e., matching pairs with no interactions, are not shown for the sake of clarity

where $K\mathbf{x}^m = \frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}^m}$. In [9], $K\mathbf{x}^m$ leads to relaxing the QAP to a linear assignment problem. It can be interpreted as the suitability of $\mathbf{x}^m$ for becoming the optimal matching. More precisely, each column of $K\mathbf{x}^m$ pools the confidences associated with a candidate match [4]. As a result, the relevance of a matching ik is given by:

$$\mathbf{I}_{DP} = (K\mathbf{x}^m)_{ik} = \sum_{jl} \mathbf{x}^m_{jl} K_{ikjl} \tag{10}$$

This expression is considered a *sum pooling* since for any candidate match ik it accumulates all the affinities leading to close a rectangle, which rather differences from the MF strategy that accounts for all *potential rectangles*, regardless of their weight. When used for Graph Matching, as in [4], Eq. 13 is a *weighted* sum pooling since the matching variables $\mathbf{x}^m_{ik} \in [0, 1]$.

$$\mathbf{I}_{DP}(a) = (K\mathbf{x}^m)_{ik} = \mathbf{x}^m_{ik} K_{iikk} + \sum_{ik}\sum_{jl} \mathbf{x}^m_{jl} K_{ikjl} \tag{11}$$

Consequently, the weighted sum pooling is prone to noise. However, in a m-best context, where $\mathbf{x}^m_{ik} \in \{0, 1\}$, the mode $\mathbf{x}^m$ filters out uninformative elements (i.e. noisy scores) and $K\mathbf{x}^m$ smoothes the search space, providing a nice trade-off between depth search and breadth search.

Then, with $(K\mathbf{x}^m)_{ik}$ at hand, the so called *discrete pooling* (DP) selection strategy retains the following variable $v = \arg\max_{ik \in \mathcal{D}(\mathbf{x}^m)} \mathbf{I}_{DP}(ik)$. Again, for each matching ik in which i is involved we retain the k yielding the maximum TC for ik, and finally we select the variable associated with the maximum of all these maxima.

In Fig. 2, given the *affinity matrix* K (Fig. 2-Top right), we have that the relevance of a given edge iy_i, in particular $2b$, comes from the correlation $K^T_{iy_i,**}\mathbf{x}^m$ between its row and the m solution. For $2b$ we have that $K^T_{2b,**}\mathbf{x}^m = K_{2b,1a} + K_{2b,4c} + K_{2b,2b} + K_{2b,4c} = 10$. However, for $3d$ we have $K^T_{3d,**}\mathbf{x}^m = 0$. For the solution m in in Fig. 2-Top left. we have that DP assigns a relevance of 10 to v_2, whereas the relevance of v_3 is 0. Consequently DP selects v_2 because its current value (b) pools more similarities (weighted rectangles) than that of v_3 .

3.4. Median Discrete Pooling

For a given depth d, the BTP model always holds the *last temporary optimal* solution $\mathbf{x}^m$, which is associated with the tree node $\mathcal{X}^{d+2,*} = \arg\max_r\{J_\Delta(\mathbf{x}^{d+2,r})\}$, and the *best previous* solution $\mathbf{x}^b$, which is associated with the node $\mathcal{X}^{1\ldots d+1,*} = \arg\max_r\{J_\Delta(\mathbf{x}^{1\ldots d+1,r})\}$. So far, all the proposed strategies generate the next candidate solution by comparing $\mathbf{x}^m$ and $\mathbf{x}^b$. In addition, all variables v are referred to $\mathbf{x}^m$. This usually introduces a depth-first bias which may lead to over-explore low-energy (bad) subspaces, although this behaviour will depend on each variable selection strategy. This is why *median discrete pooling* MDP aims to set a trade-off between breadth-first search and depth-first search. In order to do so, we compute both $K\mathbf{x}^m$ and $K\mathbf{x}^b$.

$$\mathbf{I}_{MDP}(i) = \sum_{k_i}\sum_{jl} x^m_{jl} K_{ik_ijl} + \sum_{k_j}\sum_{jl} x^b_{jl} K_{ik_jjl} \tag{12}$$

For each matching ik, where $i \in \mathcal{D}$ (let $\mathcal{D}$ be the set of variables whose assignments are different, and consider also the solutions $\mathbf{x}^m$ and $\mathbf{x}^b$, in the example $\mathcal{D} = \{2, 3\}$) and k represents all different v_i ($k_i = v^m_i$ and $k_j = v^b_i$), we compute the median given by $M = \text{med}\{I_{MDP}(i)\}$ and the selected variable i is the one that minimizes $|I_{MDP}(i) - M|$. In this case (see Fig. 2-Bottom right) we have that $M = 5$ and thus, any variable v_2 or v_3 is equally closer (in terms of MDP) from M. This illustrates the breadth-first behaviour of MDP in contrast with DP.

Our four strategies are studied in our experiments in Section 4, evaluating and testing our hypothesis.

4. Experimental Results

Once we have studied the m-best solutions to Graph Matching, which rely on exclusion strategies where the k-th best solution is found by excluding all the k-1 better ones from the search space. The main problem is that the state-of-the-art methods are generic insofar they do not incorporate the detailed structure of the problem, e.g. the properties of the global affinity matrix which characterise the search space.

For that, in this section, we propose several strategies or heuristics for spanning the whole range between depth-first and breadth-first search for improving the state-of-the-art methods in terms of accuracy and efficiency. We follow up the experimental set performed in the test of the Hungarian BP + BTP method [23] which can be seen as the state-of-the-art technique for m-best graph matching. Herein, we focus on the car (30 pairs) and motor (20 pairs) datasets, both from the PASCAL VOC 2007 challenge [7].

The graph matching algorithms compared are: the Graduated Assignment (GA) [9], Probabilistic Matching (PM) [31], Spectral Matching (SM) [16], Spectral Matching with Affine Constraints (SMAC) [5], Integer Projected Fixed Point Solver (IPFPS) [17], Reweighted Random Walks Matching (RRWM) [3], Factorized Graph Matching (FGM) [33] and finally, BP (Hungarian BP + BTP) [23].

4.1. Experiment #1: Comparing all Strategies

In Table 1 and Fig. 3, we show the average matching accuracies for the 30 Car pairs and 20 Motor pairs as the number of outliers increases from 0 to 20. Regarding the m-best strategy, we set $m = 5$. The strategies compared are:

Table 1. Matching accuracy and processing times for Car Dataset with different number of outliers

Outliers	PM	GA	SM	SMAC	IPFPS	RRWM	FGMD	BP	$IPFPS_5$	BP_5	**MF$_5$**	**DP$_5$**	**MDP$_5$**	**TC$_5$**
0	0.57	0.62	0.73	0.79	0.83	0.88	0.91	0.92	0.87	0.92	**0.93**	0.92	0.92	0.92
1	0.54	0.57	0.69	0.75	0.78	0.86	0.89	0.89	0.82	**0.89**	0.88	0.88	0.88	**0.89**
2	0.50	0.54	0.66	0.74	0.77	0.87	0.89	0.88	0.80	0.88	0.88	**0.89**	0.88	**0.89**
3	0.46	0.54	0.65	0.74	0.76	0.83	0.85	0.87	0.80	0.89	**0.90**	0.89	0.89	0.88
4	0.47	0.52	0.67	0.72	0.76	0.84	0.87	0.85	0.79	0.85	0.85	**0.87**	0.86	0.84
5	0.40	0.50	0.61	0.70	0.77	0.81	0.86	0.87	0.80	0.87	0.88	0.88	**0.89**	**0.89**
6	0.39	0.46	0.61	0.70	0.72	0.77	0.84	0.83	0.75	0.87	0.88	**0.89**	0.86	0.86
7	0.40	0.48	0.63	0.66	0.80	0.80	0.81	0.87	0.81	0.84	0.84	**0.85**	**0.85**	0.84
8	0.41	0.47	0.57	0.67	0.69	0.78	0.77	0.83	0.69	0.86	0.85	0.86	0.85	**0.87**
9	0.39	0.44	0.56	0.65	0.67	0.76	0.79	0.83	0.68	0.82	0.82	0.82	0.81	**0.83**
10	0.36	0.42	0.54	0.64	0.70	0.76	0.80	0.82	0.71	0.84	0.84	**0.86**	0.85	0.82
11	0.36	0.41	0.53	0.66	0.67	0.75	0.82	0.80	0.70	**0.86**	0.85	**0.86**	0.85	**0.86**
12	0.34	0.41	0.57	0.66	0.70	0.77	0.76	0.77	0.72	**0.80**	**0.80**	0.79	0.79	0.79
13	0.35	0.41	0.54	0.65	0.69	0.79	0.75	0.77	0.70	0.77	0.79	0.79	0.77	**0.80**
14	0.36	0.39	0.55	0.63	0.69	0.77	0.81	0.80	0.71	0.79	0.78	0.79	**0.80**	0.79
15	0.35	0.38	0.49	0.65	0.64	0.77	0.81	0.80	0.65	0.75	**0.78**	0.76	0.77	0.77
16	0.30	0.35	0.50	0.62	0.61	0.79	0.75	0.74	0.62	0.79	0.79	**0.81**	0.80	0.79
17	0.34	0.37	0.51	0.65	0.63	0.75	0.79	0.76	0.63	0.79	0.78	**0.81**	0.79	0.79
18	0.31	0.35	0.47	0.61	0.59	0.73	0.76	0.78	0.60	0.76	0.75	**0.81**	0.77	0.77
19	0.29	0.34	0.50	0.62	0.66	0.70	0.68	0.76	0.67	0.78	**0.80**	**0.80**	0.79	0.77
20	0.29	0.33	0.46	0.57	0.60	0.70	0.65	0.69	0.60	0.75	**0.77**	**0.77**	0.75	0.76
Time	0.01	0.03	0.03	0.03	0.07	1.38	19.18	4.90	0.63	264.54	259.89	**253.86**	256.76	258.67

- **BP (Hungarian BP + BTP)** [23]. This strategy is based on the BP Graph Matching method [32]. This graph-matching approach is the state-of-the-art, since the Binary Tree Partition (BTP) strategy must be initialised with a high-quality MAP solution. Even with this starting solution is provided, the performance of the BP method

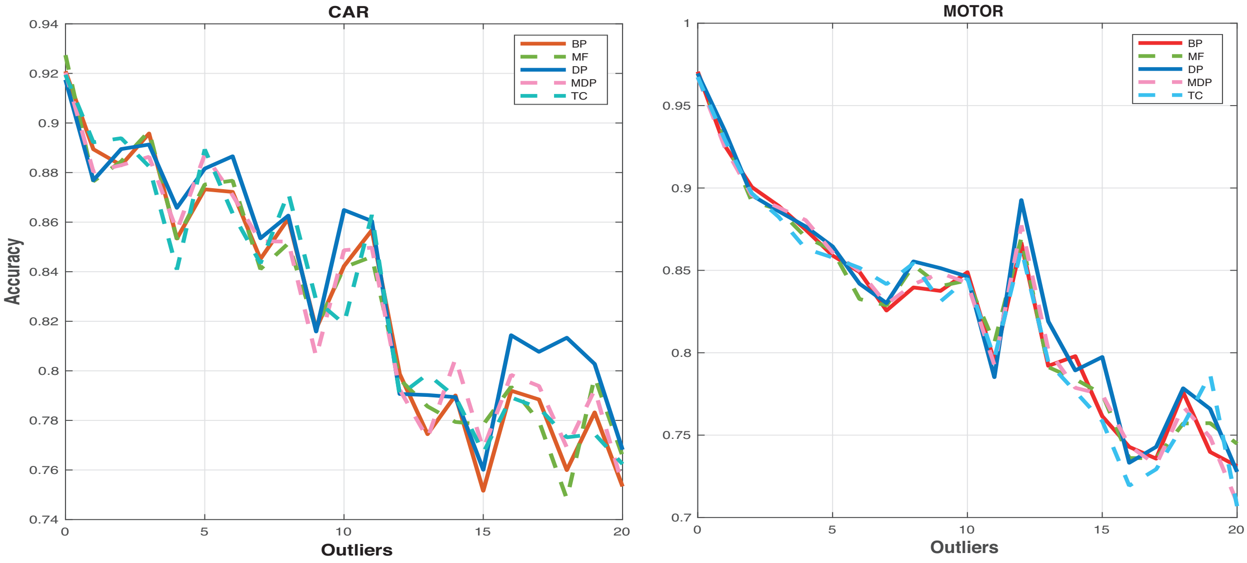


Figure 3. Accuracy of the alternative strategies for Car dataset

degrades as the number of outliers increases.

- **MF (Max Factor)**. However, the MF variable selection exploits the structure of the problem by accounting for the *potential* number of rectangles that can be closed by a given matching. For a moderate number of outliers, MF slightly improves the accuracy (1% for the Car dataset and similar values for the Motor dataset). However, it also degrades for more than 11 outliers. A detailed analysis by pairs reveals that certain matching problems (pairs 4, 5, 8, 25 and 28) are very hard to solve for all the Graph Matching methods. The affinity matrices for these pairs are characterised by spectra with uneven inter-eigenvalue gaps. These pairs are penalised by the depth-first behaviour of BTP and the MF selection cannot recover from a wrong decision for small values of m. However, in these pairs the random variable selection provides a trade-off between breadth-first and depth-first. For the remaining pairs MF improves BP. As we will detail in experiment #3 in Subsection 4.3, MF selection requires smaller values of m to improve the accuracy of BP.

- **DP (Discrete Pooling)**. The DP variable selection improves both BP and MF even for a moderate number of outliers. This is due to the filtering provided by the discrete accumulation of scores. In DP, only existing rectangles are considered whereas MF accounts for all potential rectangles. This selective behaviour flattens the search space, and this improves the breadth-first strategy of BP. DP tends to explore a subspace in depth when it is sure that it is promising. In addition, the tolerance to outliers and the selective behaviour yield better increments of accuracy as m increases. For instance, with $m = 10$ and 20 outliers, DP significantly outperforms BP (4%, see Fig. 5 and Table 2)

- **MDP (Median Discrete Pooling)**. The main difference between Median DP and DP is that MDP seeks for diversity (breadth-first). When MDP outperforms DP, the gain is quite significant. However, MDP requires a large number of modes to capture the most promising subspace.

- **TC (Trimmed Centrality)**. TC (with p=6 eigenvectors) works well only for a moderate number of outliers. It slightly improves the accuracy of BP, but as MF its performance degrades for *difficult pairs*. It slightly relaxes the depth-first behaviour of BP, but it is the more depth strategy (see experiment 4.3).

Moreover, we use the CMU house image dataset, that consists of 111 frames of a house, each of which has been manually labelled with 30 landmarks. We show a performance using 20 nodes, because the total performance produces a perfect matching in the state of the art. We matched all possible image pairs, spaced by 0:10:90 frames and computed the average matching accuracy per sequence gap. In Fig. 4.1 we can see that our algorithms improve BP (1 %), but the improvement is less than in Car and Motor datasets due to the difficult of pairs implied in matching.

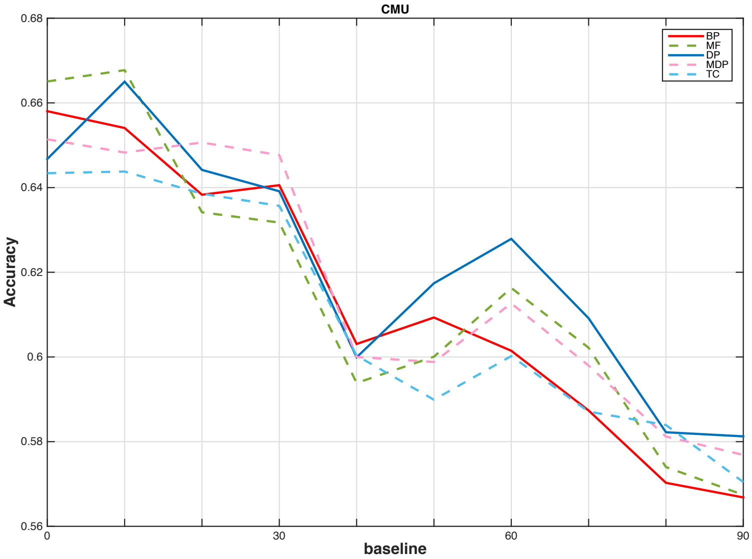


Figure 4. Comparison of all algorithms on the CMU house datasets using 20 nodes

We complement these experiments (done for $m = 5$) with an asymptotic test with $m = 10$. This analysis is motivated by the results obtained for $m = 5$, where we found diverging accuracies for different pairs. For the Car Dataset we identified several pairs as *difficult* or *hard* (those with a matching accuracy below 75% for BP). These pairs are: 2, 4, 5, 8, 24, 25, 26, 28 and 29 (see pairs in red in Table 5). This threshold (75%) allows us to study pairs with a certain degree of confusion/ambiguity. We have that MF and TC are outperformed by BP in these pairs but DP and MDP improve BP therein. In addition, DP holds its robustness in these pairs.

Summarising the analysis for $m = 10$, DP is the winning strategy for difficult pairs and it is also the most robust strategy (high tolerance to outliers as we can see in Fig. 5). This complements Table 2, where the data is shown according to an increasing number of outliers (from 0 to 20). However, what is the expected behaviour when m increases (better marginalisation)?. We note that DP can outperform significantly BP for a large value of m.

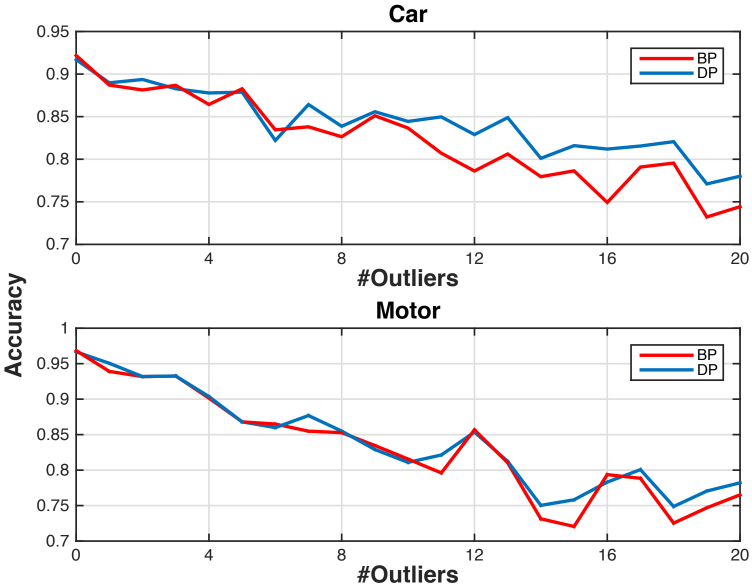


Figure 5. The accuracy for Car (left) and Motor datasets (right) as the number of outliers increases ($m = 10$).

Rezatofighi et al. [23] hold that as m increases the probability of finding the best solution (ideally to guess the ground truth) is higher. However, this statement is only true when each discovered mode is among the best possible modes.

In experiment #3 (Figures 8 and 9, Tables 3 and 4) is targeted to analyze the quality of the modes obtained by all the alternatives.
In practice, we have that the asymptotic accuracy depends on the quality of the initial modes ($m = 1, 2, \ldots$). A bad choice in $m = 1$ often leads to a depth search. Therefore, any strategy needs a large value of m to recover from these failures. This fact is critical when the number of outliers increases. In particular, we want to be sure that our improvement over the random selection strategy (BP) still holds for $m > 5$. In other words, we run BP and DP for $m = 10$ to favour the recovery of BP.

In Table 2 (see also Figure 5), we show both the matching accuracy and running time for BP and DP when $m = 10$. We observe that for $m = 10$, DP is even more robust and better conditioned for dealing with hard pairs, than in the $m = 5$ case. This is motivated by the fact that DP has the chance of being more diverse (flattening the search space) as m increases. For hard pairs in the Car dataset, we have an improvement of 2% for $m = 5$, whereas for $m = 10$ the improvement peaks 4.5%. The average accuracy of DP for $m = 10$ is 84.33% whereas that of BP is 82.32%. The average running time of DP for $m = 10$ is 720.02 seconds, whereas that of BP is 759.96 seconds. However, for $m = 5$ the average accuracies for DP and BP are 83.87% and 82.82% respectively, and the average running times are 253.86 (DP) vs 264.54 seconds (BP). In the Motor dataset, we have less difficult pairs (only pairs 14 and 15), and obtain an improvement of 2.5% in $m = 5$, whereas the average accuracies are 83.26 (DP) and 82.26 (BP), and running time are 188.77 (DP) and 197.4 (BP) seconds, respectively. In Fig. 6, we show how BP and DP match the cars in Pair 8 (without outliers and $m = 10$).

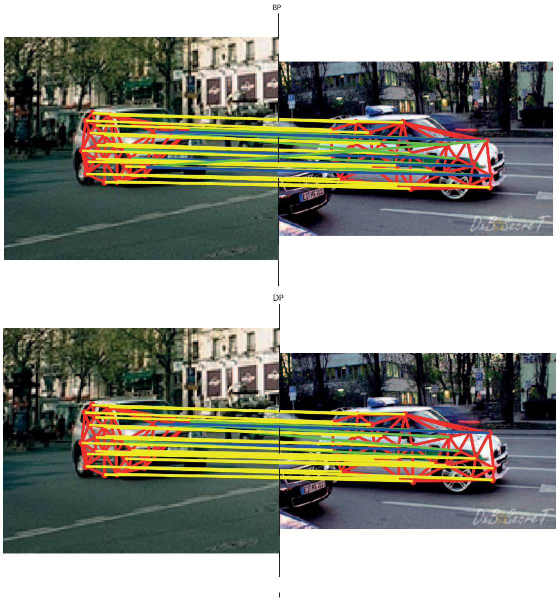


Figure 6. Matching of Pair 8 for Car Dataset without outliers. Top-BP with 76.92% of accuracy and Botton-DP with 84.62%. False positive matchings in green, and false negative matchings in blue.

4.2. Experiment #2: Exploring Solutions Subspaces for m=100

In previous experiments, we show compare our strategies with the state-of-the-art with the same number of m=5, to keep coherence with previous works [23]. We obtain better results in terms of accuracy and efficiency. As we explain in Subsection 2.4, and we have illustrate in Fig. 1 and algorithm 1, the two initial modes are the same for all strategies. From third solution, the variable selection strategies are relevant to a correct exploration of the search space of solutions to reach the optimal solution. In particular, in difficult pairs is critical the selection of a balanced strategy that can be able to recover from wrong solutions, as we can see in Table 5. Our hypothesis is that the choice of the number of m is not the most important point in this paper, because the main goal is to provide a criterion to find the best solution with less number of m, characterizing the search space of solutions.

For that, we propose the following experiment: we explore the long-term behaviour ($m = 100$) of all the strategies for a *difficult pair* (Pair 8 of Car dataset). We consider 0, 10 and 20 outliers. Our aim is to study the correlation between the accuracy and the value of the objective function as the number of solutions increase. We also analyse the jumping trends of each strategy. In Fig. 4.2, we show that for a low number of outliers, all the strategies behave similarly but reaching best accuracy earlier than BP. As we have seen in Experiment #1, both DP and MPD tolerate a high number of outliers. Their Hamming pairwise distances show that pooling methods a quite focused in a limited number of subspaces (see blocks in Figure 4.2).

In Subsection 4.1, we explain that in difficult pairs, as we show in this experiment, BP searches solutions in the search space of solutions in a randomly way (diversity), and needs a high number of solutions to recover from failures. In contrast, DP need not a high number of m to obtain the best result because it recovers better from failures. This explanation is complemented with an experiment with m=10 (Table 2) and in with this experiment, where we show the curves of accuracy and objective value of a difficult pair with m=100. DP reaches the best result with a less number of m (i.e. without outliers, BP reaches its best result with m=11, DP reaches its best case in with m=4). Moreover, we show that our strategies works very well with a high number of outliers.

Table 2. Matching accuracy and processing times of BP and DP for all pairs for Car and Motor Datasets with different number of outliers with m=10

Outliers	BP_{car}	DP_{car}	BP_{motor}	DP_{motor}
0	0.92	0.92	0.97	0.97
1	0.89	0.89	0.94	0.95
2	0.88	0.89	0.93	0.93
3	0.89	0.88	0.93	0.93
4	0.86	0.88	0.90	0.90
5	0.88	0.88	0.87	0.87
6	0.83	0.82	0.86	0.86
7	0.84	0.86	0.85	0.88
8	0.83	0.84	0.85	0.85
9	0.85	0.86	0.83	0.83
10	0.84	0.85	0.82	0.81
11	0.81	0.85	0.80	0.82
12	0.79	0.83	0.86	0.85
13	0.81	0.85	0.81	0.81
14	0.78	0.80	0.73	0.75
15	0.79	0.82	0.72	0.76
16	0.75	0.81	0.79	0.78
17	0.79	0.82	0.79	0.80
18	0.80	0.82	0.73	0.75
19	0.73	0.77	0.75	0.77
20	0.74	0.78	0.77	0.78
Avg Acc	0.82	**0.84**	0.83	**0.84**
Avg Time	759.96	**720.02**	480.23	**448.26**

4.3. Experiment #3: Test of Efficiency Dual and Dual-Counterpart

In this third experiment, our aim is to estimate the average number of modes needed to outperform the alternatives, for $m = 5$. In other words, we test the efficiency in terms of *how many solutions are needed to outperform the state of the art (BP)*. In the Dual case we evaluate how many number of modes are needed to outperform BP. If it is needed more than 10 solutions, we stop in $m = 10$ (yellow bars in Figures 8 and 9). As we show in Tables 3 and 4, our proposed variable selectors need only around 37% of the total number of modes to outperform BP in both datasets (i.e. in dual test of car dataset, BP needs 3150 solutions to obtain a 83.31% whereas DP needs only 1171 solutions to obtain a 84.67% of accuracy). In the Dual-Counterpart case we fix $m = 5$ for DP and evaluate the number of modes needed by the alternatives to outperform our best strategy (DP). In this case, the alternatives require close to 45% of the total number of modes to outperform DP in car dataset (1394 solutions), and 46% in motor dataset. The global improvement of DP is a 8% in car dataset and 9.04% in motor dataset.This is due to an early stop of DP with respect to the alternatives as we can see in Fig. 8 and Fig. 9, where in dual test (left) we need fewer solutions to outperform to BP (blue bars) in contrast of dual counterpart test (right), where we need more solutions (smaller blue bars). We also show therein that in the Dual-Counterpart, the alternative strategies often require $m = 10$ or more solutions to outperform DP (higher yellow bars). In running time, DP improves a 36.45% (car) and 15.15% (motor) of global improvement to BP (i.e., in car dataset, dual test has 267.93 sec vs 87.67 sec, and in dual counterpart test we have 242.29 sec vs 165.84 sec). In conclusion, *DP is the best strategy both in terms of the number of modes and in running time*.

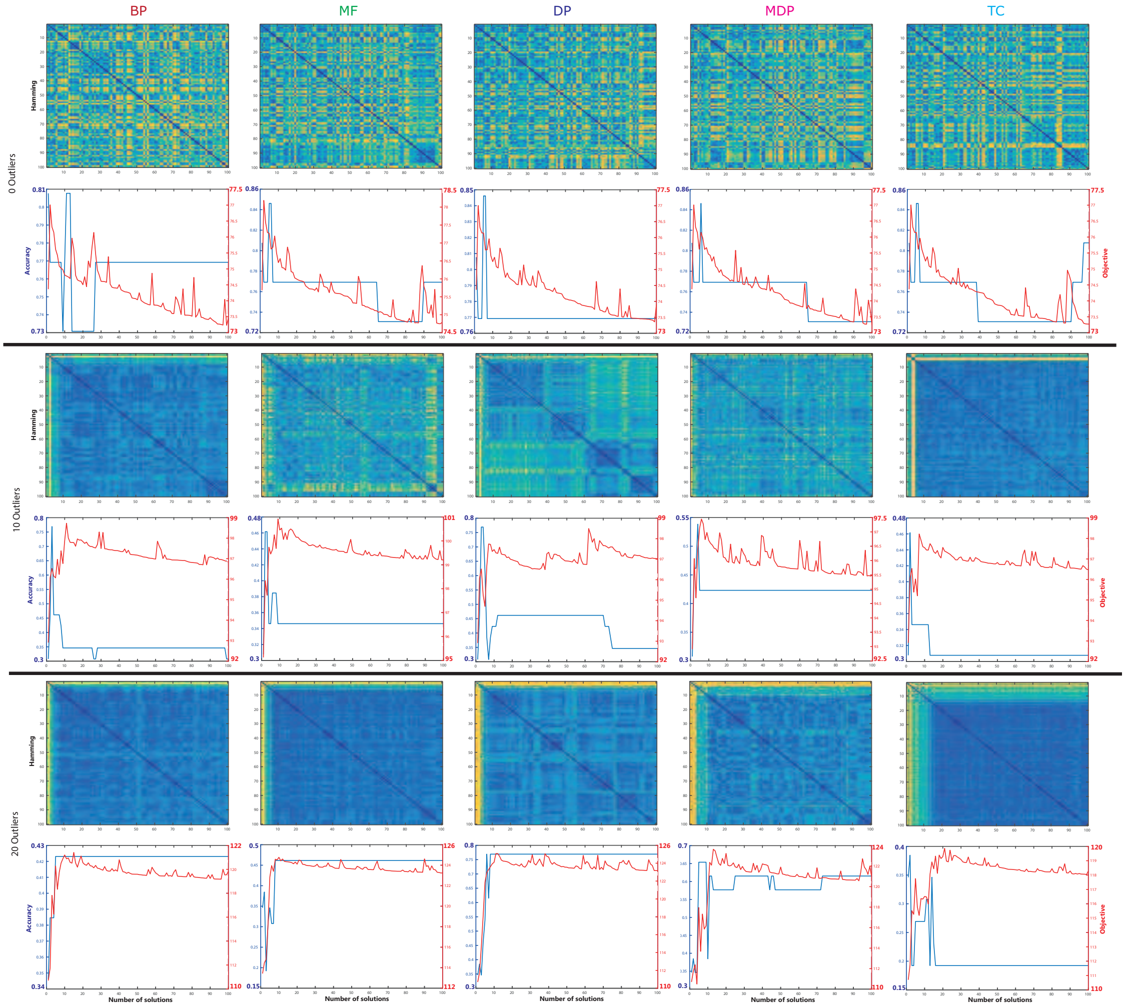


Figure 7. Exploring the long-term behaviour (100 m-best iterations). We show a difficult pair, number 8, representing the Hamming pairwise distances and Accuracy vs Objective for different number of outliers.

Table 3. Dual (left) and Dual Counterpart (right) for Car Dataset (3150 solutions are the accumulative number of m solutions from 30 (pairs) × 21 (cases of outliers, from 0 to 20) × 5 (m-best))

Alg	Avg Acc	Avg Time	M Times	Alg	Avg Acc	Avg Time	M Times
BP	0.8331	267.93	**3150**	DP	0.8428	242.29	**3150**
MF	0.8441	106.04	1221	BP	0.8488	165.94	1394
DP	0.8467	87.67	1171	MF	0.8507	147.97	1337
MDP	0.8450	98.08	1256	MDP	0.8474	119.67	1409
TC	0.8434	96.10	1259	TC	0.8476	127.50	1402

Table 4. Dual (left) and Dual Counterpart (right) for Motor Dataset (2100 solutions are the accumulative number of m solutions from 20 (pairs) × 21 (cases of outliers, from 0 to 20) × 5 (m-best))

Alg	Avg Acc	Avg Time	M Times	Alg	Avg Acc	Avg Time	M Times
BP	0.8319	192.24	**2100**	DP	0.8404	184.03	**2100**
MF	0.8497	110.47	827	BP	0.8446	94.38	991
DP	0.8496	70.04	803	MF	0.8469	124.80	907
MDP	0.8492	72.76	827	MDP	0.8451	88.02	953
TC	0.8490	70.80	809	TC	0.8461	92.09	954

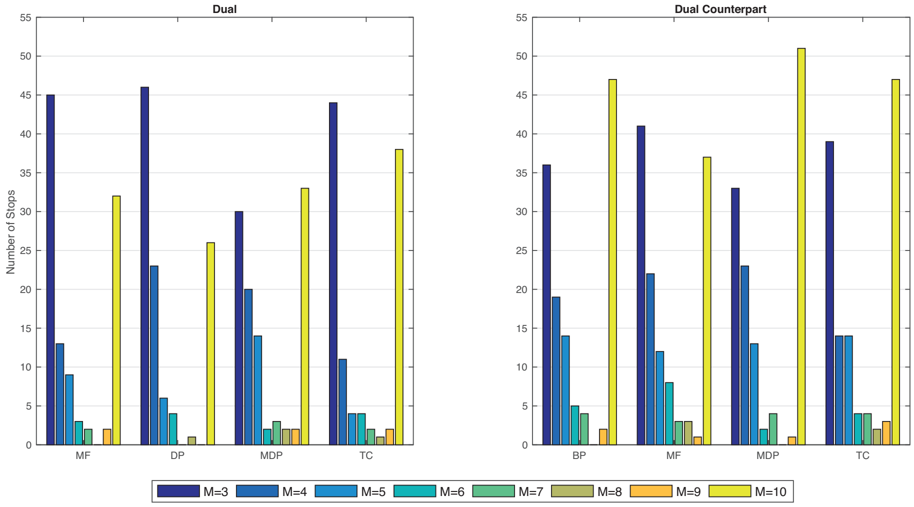


Figure 8. Dual and dual counterpart for Car Dataset

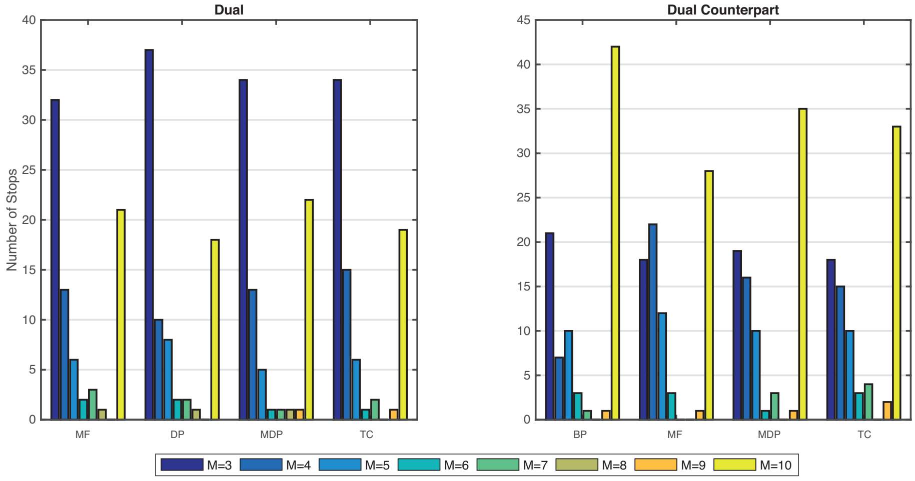


Figure 9. Dual and dual counterpart for Motor Dataset

4.4. *Experiment #4: Analysis by pairs by spectral characterization*

We have often referred to both *easy* and *hard* pairs. In general, the difficulty of a graph matching problem can be determined by the degree of indefiniteness of the matrix $\mathbf{K}$. However, in practice we often face similar degrees of indefiniteness (see Fig. 11-Top). Consequently, we need a more precise characterization of the difficulty of a graph matching problem.

Table 5. Matching accuracy and processing times of all pairs for Car Dataset with different number of outliers for all strategies

Pair	1	2	3	4	5	6	7	8	9	10
BP	0.76	0.69	0.96	0.30	0.71	0.87	0.99	0.51	1.00	0.93
MF	0.77	0.71	0.96	0.29	0.73	0.87	0.99	0.48	1.00	0.93
DP	0.78	0.70	0.97	0.33	0.74	0.88	0.99	0.54	1.00	0.94
MDP	0.77	0.71	0.96	0.31	0.67	0.86	0.99	0.52	1.00	0.92
TC	0.77	0.71	0.96	0.31	0.65	0.87	0.99	0.52	1.00	0.94

Pair	11	12	13	14	15	16	17	18	19	20
BP	0.99	0.97	0.98	0.89	0.93	0.96	0.85	0.97	0.98	0.97
MF	0.99	0.97	0.98	0.88	0.92	0.96	0.87	0.97	0.98	0.97
DP	0.99	0.97	0.98	0.88	0.92	0.97	0.90	0.97	0.97	0.97
MDP	0.99	0.98	0.99	0.89	0.92	0.97	0.85	0.96	0.97	0.97
TC	0.99	0.97	0.98	0.89	0.92	0.97	0.87	0.97	0.98	0.97

Pair	21	22	23	24	25	26	27	28	29	30
BP	0.99	0.90	0.82	0.70	0.55	0.71	0.97	0.45	0.75	0.81
MF	0.99	0.91	0.82	0.71	0.50	0.71	0.97	0.49	0.76	0.80
DP	0.99	0.90	0.83	0.71	0.58	0.72	0.97	0.48	0.77	0.81
MDP	0.99	0.91	0.83	0.76	0.53	0.73	0.97	0.47	0.73	0.81
TC	0.99	0.88	0.82	0.74	0.53	0.71	0.97	0.49	0.76	0.80

4.5. *Discrete Pooling flattens the search space: Difficult Problems*

Our experiments with $m = 5$ show that Discrete Pooling is, by far, the best strategy for variable selection. As we note in the paper, this is consistent with the results obtained in [4], where discrete pooling is used as a means of tolerating a large number of outliers. However, in an m-Best Graph-Matching context, we have a) to exploit the fact that the matching variables ($\mathbf{x}^m_{ik}$ and $\mathbf{x}^m_{jl}$) in

$$\mathbf{I}_{DP}(a) = (\mathbf{K}\mathbf{x}^m)_{ik} = \mathbf{x}^m_{ik}\mathbf{K}_{iikk} + \sum_{ik}\sum_{jl}\mathbf{x}^m_{jl}\mathbf{K}_{ikjl} \tag{13}$$

are *discrete*, and b) analyse the role of $\mathbf{K}\mathbf{x}^m$ (the derivative of the cost function $\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}^m}$) at $\mathbf{x}^m$) as a means of quantifying the local gradient of the cost function. In this regard, the existence of both node and edge attributes in $\mathbf{K}$ transforms a locally flat landscape into a more rugged one, which is dominated by low-frequency modes. We are interested in hopping between high-frequency modes within a given low-frequency one, ideally the mode containing the best solutions. Therefore, we have to ensure that we are not going to jump to another low-frequency mode unless it is required.

Since DP is embedded in a Binary Tree Partition structure, which is depth-first focused, the risk of jumping to another low-frequency mode is very high. Why does DP find an intelligent trade-off between depth-first and breadth-first search?

In order to answer this question we have complemented **Experiment #1** with the following test. Given an affinity matrix $\mathbf{K}_e$ supposed to characterise an easy graph matching problem (for instance Pair 1 in the Car Dataset) we have slightly perturbed the ground truth (the diagonal). Let $\mathcal{S}_n$ the set of $\binom{n}{2}$ pairs (i, j), where n is the number of features to match. Then, $\mathcal{S}_n = \{(1,2),\ldots,(1,n),(2,3),\ldots,(2,n),\ldots,(k,k+1),\ldots,(k,n)\ldots(n-1,n)\}$. For each element of $(i, j) \in \mathcal{S}_n$ we have replaced the matching $i \to i$ (ground truth) by $i \to j$ while leaving the remaining matchings unchanged. In this way, we produce $\binom{n}{2}$ perturbations of $\mathbf{x}$ for the same $\mathbf{K}$. These perturbations simulate an energy landscape of $\binom{n}{2}$ modes for $\mathbf{xKx}$.

In Fig. 10 we plot $J(\mathbf{x}) = \mathbf{x}^T\mathbf{Kx}$ vs $J(\mathbf{x}) + \max(\mathbf{Kx})$ (left column). For the first and second row we have an easy pair with 0 and 10 outliers respectively. The local ruggedness of the energy landscape increases with the number of outliers (i.e. with the degree of ambiguity). Let then $\Delta = \max(\mathbf{Kx})$ the maximum derivative (it is actually the value used for variable selection). It is worth noting that this upper bound of the local derivative flattens the neighbouring discrete poolings (see the right columns of Fig. 10) in such a way that DP increases the diversity of the search space. The two first rows correspond to the easy case ($\mathbf{K}_e$) with 0 and 10 outliers respectively. Therefore, as the number of outliers increases, the high-frequency modes get closer and can be more easily smoothed by Δ. In addition, the smoothing effect of Δ is stronger for a *hard pair* ($\mathbf{K}_h$), such as Pair 5 in Car Dataset, which is populated with high-frequency modes even for 0 outliers (third row in Fig. 10). There are no significant differences in terms of Δ for the same hard pair and 10 outliers.

As a conclusion, *DP increases the diversity of the search space (the breadth-first behaviour) while keeping the correct searching direction*. In terms of the associated Partitioning Enumeration Subspaces (PES), DP finds the correct subspace earlier than the alternatives. This behaviour is held when the number of outliers increases, because this regime produces deeper BTPs and an intelligent trade-off between depth-first search and breadth-first search is critical.

5. Conclusions and Future Work

In this paper, we have characterized the search space of QAP in m-best Graph Matching problems. We transform a MAP problem into a m-best one where we combine both approximation and marginalization strategies. We propose four new different strategies to discuss the randomly criterion of variable selection of the state of the art. To that end, we have proposed four strategies for selecting the next variable to clamp (MF, TC, DP and MDP). The experiments show that all the proposed strategies reaches important improvements, outperforming the state-of-the-art (random variable selection or BP) in terms of matching accuracy, being DP the best one (from 1% with m=5 until 2% with m=10), specially when the level of noise increases (outliers). Moreover, the experiments performed with difficult matching pairs show that the improvement reaches 4.5%. Concerning efficiency, all the proposed heuristics speed-up the process, up to 5% on average with respect to BP.

Regarding the exploration of the search space, BP, MF and TC are more sensitive to errors (selection of wrong variables) whereas DP and MDP provide a trade-off between breadth-based search (diversity) and depth-based (digging in a promising subspace). As a result, DP and MDP can recover earlier from wrong variable selections. Our experiments show that the most robust heuristic is DP. Finally, the amount of exploration needed for finding the optimal solution, all our strategies improve BP. Here, DP (the best, again) needs only to explore 37% of the solutions (modes) analyzed by BP (for m=5) to outperform it. Conversely, BP needs to explore 45% of the solutions analyzed by DP for improving it. Therefore, DP is the best heuristic concerning the relative exploration effort.

Summarising, we show how the choice of variable selection strategy in m-best matching can improve the performance of the state-of-the-art criterion (random selection) in terms of both accuracy and efficiency. This study opens up the possibility for finding the optimal solution with a small numbers of iterations through a deeper study of the structure of the search space when an appropriate variable selection strategy is used.

References

[1] Batra, D., 2012. An efficient message-passing algorithm for the M-best MAP problem, in: Proc. UAI, pp. 121–130.

[2] Batra, D., Yadollahpour, P., Guzmán-Rivera, A., Shakhnarovich, G., 2012. Diverse M-best solutions in markov random fields, in: Proc. ECCV, pp. 1–16.

[3] Cho, M., Lee, J., Lee, K., 2010. Reweighted random walks for graph matching, in: Proc. ECCV.

[4] Cho, M., Sun, J., Duchenne, O., Ponce, J., 2014. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers, in: Proc. CVPR, pp. 2091–2098.

[5] Cour, T., Srinivasan, P., Shi, J., 2006. Balanced graph matching, in: Proc. NIPS, pp. 313–320.

[6] Estrada, E., Rodríguez-Velázquez, J.A., 2005. Subgraph centrality in complex networks. Phys. Rev. E 71, 056103.

[7] Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A., . The pascal visual objects classes challenge 2007 (VOC 2007) results.

[8] Fromer, M., Globerson, A., 2009. An lp view of the m-best map problem, in: Proc. NIPS.

[9] Gold, S., A., R., 1996. A graduated assignment algorithm for graph matching. IEEE Trans. Pattern Anal. Mach. Intell. 18, 377–388.

[10] Hu, N., Rustamov, R.M., Guibas, L., 2013. Graph matching with anchor nodes: A learning approach, in: Proc. CVPR, pp. 2906–2913.

[11] Kirillov, A., Savchynskyy, B., Schlesinger, D., Vetrov, D.P., Rother, C., 2015a. Inferring m-best diverse labelings in a single one, in: Proc. ICCV, pp. 1814–1822.

[12] Kirillov, A., Shekhovtsov, A., Rother, C., Savchynskyy, B., 2016. Joint M-best-diverse labelings as a parametric submodular minimization. CoRR (accepted in NIPS'2016) abs/1606.07015.

[13] Kirillov, A., Shlezinger, D., Vetrov, D.P., Rother, C., Savchynskyy, B., 2015b. M-best-diverse labelings for submodular energies and beyond, in: Proc. NIPS, pp. 613–621.

[14] Lawler, E.L., 1963. The quadratic assignment problem. Management science 9, 586–599.

[15] Lawler, E.L., 1998. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. Management Science 8, 159–173.

[16] Leordeanu, M., Hebert, M., 2005. A spectral technique for correspondence problems using pairwise constraints, in: Proc. ICCV, pp. 1482–1489.

[17] Leordeanu, M., Sukthankar, R., Hebert, M., 2012. Unsupervised learning for graph matching. International journal of computer vision 96, 28–45.

[18] Lyzinski, V., Fishkind, D.E., Fiori, M., Vogelstein, J.T., Priebe, C.E., Sapiro, G., 2016. Graph matching: Relax at your own risk. IEEE Trans. Pattern Anal. Mach. Intell. 38, 60–73.

[19] Meltzer, T., Yanover, C., Weiss, Y., 2005. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation, in: Proc. ICCV), 17-20 October 2005, Beijing, China, pp. 428–435.

[20] Nilsson, D., 1998. An efficient algorithm for finding the M most probable confgurations in probabilistic expert systems. Statistics and Computing 8, 159–173.

[21] Park, D., Ramanan, D., 2011. N-best maximal decoders for part models, in: Proc. ICCV, pp. 2627–2634.

[22] Ramakrishna, V., Batra, D., 2012. Mode-marginals: Expressing uncertainty via diverse m-best solutions., in: NIPS Workshop on Perturbations, Optimization and Statistics.

[23] Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I., 2016. Joint probabilistic matching using m-best solutions, in: Proc. CVPR.

[24] Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A.R., Reid, I.D., 2015. Joint probabilistic data association revisited, in: Proc. ICCV, pp. 3047–3055.

[25] Schellewald, C., Schnörr, C., 2005. Probabilistic subgraph matching based on convex relaxation, in: Proc. EMMCVPR, pp. 171–186.

[26] Sun, Q., Batra, D., 2015. Submodboxes: Near-optimal search for a set of diverse object proposals, in: Proc. NIPS, pp. 1378–1386.

[27] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F., Rother, C., 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1068–1080.

[28] Vogelstein, J.T., Conroy, J.M., Lyzinski, V., Podrazik, L.J., Kratzer, S.G., Harley, E.T., Fishkind, D.E., Vogelstein, R.J., Priebe, C.E., 2015. Fast approximate quadratic programming for graph matching. PLOS ONE 10.

[29] Yanover, C., Weiss, Y., 2003. Finding the M most probable configurations in arbitrary graphical models, in: Proc. NIPS, pp. 289–296.

[30] Zaslavskiy, M., Bach, F.R., Vert, J., 2009. A path following algorithm for the graph matching problem. IEEE Trans. Pattern Anal. Mach. Intell. 31, 2227–2242.

[31] Zass, R., Shashua, A., 2008. Probabilistic graph and hypergraph matching, in: Proc. CVPR.

[32] Zhang, Z., Shi, Q., McAuley, J., W., W., Zhang, Y., van den Hengel, A., 2016. Pairwise matching through Max-Weight bipartite belief propagation, in: Proc. CVPR.

[33] Zhou, F., la Torre, F.D., 2016. Factorized graph matching. IEEE Trans. Pattern Anal. Mach. Intell. 38, 1774–1789.

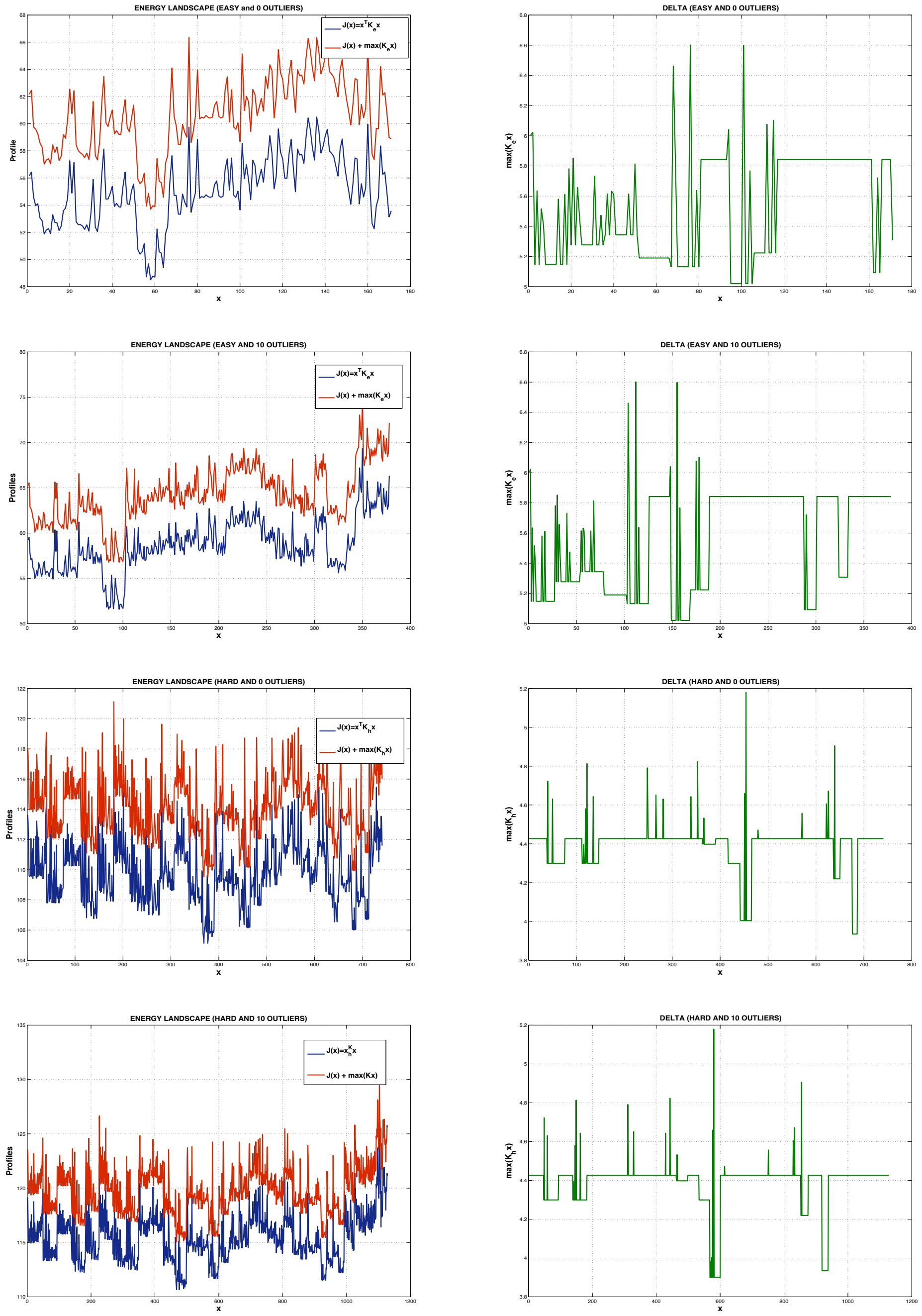
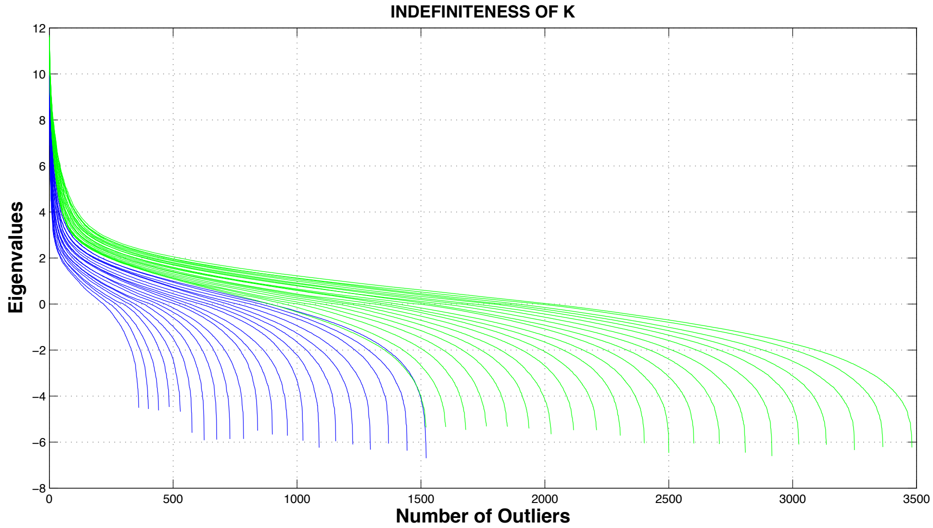


Figure 10. DP flattening the search space. First row: $J(\mathbf{x}) = \mathbf{x}^T\mathbf{K}_e\mathbf{x}$ vs $J(\mathbf{x}) + \max(\mathbf{K}_e\mathbf{x})$ for an *easy pair* (Pair 1) and 0 outliers (left) vs $\Delta = \max(\mathbf{K}_e\mathbf{x})$ (right). Second row: the same for an *easy pair* (Pair 1) and 10 outliers (left) vs $\Delta = \max(\mathbf{K}_e\mathbf{x})$ (right). Third and fourth row: same as first and second rows for a *hard pair* (Pair 5) and 0 and 10 outliers.



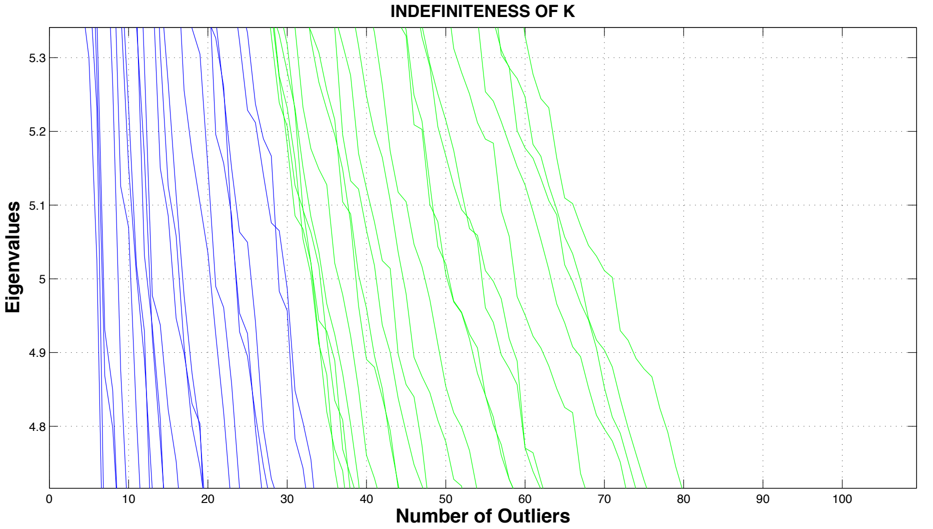


Figure 11. Spectral Characterisation of **K**. Top, the spectra of an easy pair (Pair 1) as the number of outliers increases (in blue) vs the spectra of a hard pair (Pair 5) (in green). Down: detail when the eigenvalues are large enough.