Algorithmic Operations Research Vol.5 (2010) 49–64

Probabilistic optimization in graph-problems

Cécile Murat and Vangelis Th. Paschos

LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France

Abstract

We study a probabilistic optimization model for graph-problems under vertex-uncertainty. We assume that any vertex v_i of the input-graph $G(V, E)$ has only a probability p_i to be present in the final graph to be optimized (i.e., the final instance for the problem tackled will be only a sub-graph of the initial graph). Under this model, the original "deterministic" problem gives rise to a new (deterministic) problem on the same input-graph G, having the same set of feasible solutions as the former one, but its objective function can be very different from the original one, the set of its optimal solutions too. Moreover, this objective function is a sum of $2^{|V|}$ terms; hence, its computation is not immediately polynomial. We give sufficient conditions for large classes of graph-problems under which objective functions of their probabilistic counterparts are polynomially computable and optimal solutions are well-characterized. Finally, we apply these general results to natural and well-known combinatorial problems that belong to the classes considered.

Key words: graph, complexity, approximation, a priori optimization

1. Introduction

Very often people has to make decisions under several degrees of uncertainty, i.e., when only probabilistic information about the future is available. Acquisition and validation of input data is one of the most challenging issues in almost any real-world application of operations research techniques. Although several well established theoretical models exist for problems arising in practical applications, direct application of theoretical developments may be difficult or even impossible due to incompleteness of data, or due to their questionable validity. Occasionally, one may be asked to produce an optimal operational design even before a complete deterministic picture of input data is provided, but only based on estimations and statistical measures.

We deal in this paper with the following probabilistic combinatorial optimization model under data uncertainty. Consider a generic instance I of a combinatorial optimization problem Π. Assume that Π is not to be necessarily solved on the whole I, but rather on a (unknown a priori) sub-instance $I' \subset I$. Suppose that any datum d_i in the data-set describing I has a probability p_i, indicating how d_i is likely to be present in the final sub-instance I'. Consider finally that once I' is specified, the solver has no opportunity to solve it directly (for example she/he has to react quasi-immediately, so no sufficient time is given to her/him).

In this case, a possible way for a decision maker to proceed is to compute an *anticipatory solution* S for Π, i.e., a solution for the entire instance I, and once I' becomes known, to modify S in order to get a solution S' fitting I'. The objective is to determine an initial solution S for I such that, for any sub-instance $I' \subseteq I$ presented for optimization, the solution S' respects some pre-defined quality criterion (optimality, achievement of a "good" approximation ratio, etc.).

2. Preliminaries

In what follows, we restrict ourselves in problems defined on graphs. Consider a graph $G(V, E)$ of order n, instance of a combinatorial optimization problem Π, and an n-vector $\mathbf{Pr} = (p_1, \ldots, p_n)$ of vertex-probabilities, each p_i, measuring how likely is for vertex $v_i \in V$, $i = 1, \ldots, n$, to be present in the final

---

* Part of this work has been performed while the second author was with the University of Athens on a visiting position supported by the Greek Ministry of Education and Research under the project PYTHAGORAS II

Email: Cécile Murat [murat@lamsade.dauphine.fr], Vangelis Th. Paschos [paschos@lamsade.dauphine.fr].

subgraph $G' \subseteq G$, on which the problem will be really solved. Consider a strategy $\mathtt{M}$, called *modification strategy*, such that, when a set $V' \subseteq V$ is finally realized, $\mathtt{M}$ modifies any solution S for Π into a solution S' feasible for Π in the subgraph $G' = G[V']$ of G induced by V'. Denote by $m(G', S', \mathtt{M})$ the objective value of S' in G'. Then, the value of S for G, denoted by $E(G, S, \mathtt{M})$ (and frequently called *functional*), is the expectation of $m(G', S', \mathtt{M})$, over all the possible induced subgraphs G' of G. Formally, given an anticipatory solution S, the functional $E(G, S, \mathtt{M})$ of S is defined by:

$$E(G, S, \mathtt{M}) = \sum_{V' \subseteq V} \Pr[V']\, m(G', S', \mathtt{M}) \qquad (1)$$

where $\Pr[V']$ is defined by

$$\Pr[V'] = \prod_{v_i \in V'} p_i \prod_{v_i \in V \setminus V'} (1 - p_i)$$

and represents the probability that the vertex-set V' will be the set finally present for optimization (in other words, $G[V']$ will be the instance where finally Π will be solved).

Let us note that from (1) it can be seen that the functional of a probabilistic combinatorial optimization problem is defined with respect to the chosen modification strategy $\mathtt{M}$. A different modification strategy derives a different probabilistic problem. The quantity, $E(G, S, \mathtt{M})$ can be seen as the objective function of a new combinatorial problem, derived from Π and denoted by PROBABILISTIC Π in what follows, where we are given an instance G of Π, a probability vector $\mathbf{Pr}$ on the vertices of G and a modification strategy $\mathtt{M}$. The objective is then to determine a solution S^* in G (optimal anticipatory solution) optimizing $E(G, S, \mathtt{M})$. The optimization rule of PROBABILISTIC Π is the same as the one of Π.

Concrete applications giving rise to probabilistic combinatorial optimization problems are given in [17,18]. They come from satellite shots planning, timetabling, etc. We revisit one of them, the probabilistic timetabling. Consider for a given University-fall a list of potential classes that students can follow. Any student has to choose a sublist in this list. For any class one knows the title, the lecturer and the time slot assigned to it, each such slot being proposed by the lecturer in charge. A class will open if it is chosen by sufficient students (whose the number is above a given threshold). So, nobody knows a priori if a particular class will take place before the closing of students registrations (we can reasonably assume that the choice of any student is a function of the contents of the course, of the teacher, etc.). On the other hand, one can, for example by statistical data on the behavior of the students in the past years, assign probabilities on the fact that such or such class will really open, the mandatory courses been assigned with probability 1. The problem for the University planning services is how much rooms are to be scheduled for the set of the courses offered. This problem is typically an instance of PROBABILISTIC MIN COLORING if one considers courses as vertices and if he/she links two such vertices if the corresponding classes cannot take place in the same room (because, for instance, they are planned with the same professor, or are assigned with overlapping time slots). This type of graph is known under the term "incompatibility graph". Here, an independent set, i.e., a potential color, corresponds to a set of "compatible classes", i.e., to classes that can be assigned with the same room. The number of colors used in such a graph represents the total number of rooms assigned to the set of classes considered. The probabilities resulting from the statistical analysis on the former students' behavior, are the presence probabilities for the vertices (i.e., the probabilities that the corresponding classes will really take place). Starting from an anticipatory solution, i.e., from a coloring of the incompatibility graph, the functional represents, in some sense, the average number of the necessary rooms for the courses planned.

This way to tackle data uncertainty in combinatorial optimization is called *a priori framework for probabilistic combinatorial optimization* (this term has been introduced by [9]). Under this model, restrictive versions of routing and network-design probabilistic minimization problems (in complete graphs) have been studied in [1,3–6,9–12]. In [7], the analysis of the probabilistic minimum travelling salesman problem, originally performed in [3,9], has been revisited and refined. In [15,17,8] the minimum vertex covering and the minimum coloring are tackled, while in [13,14] probabilistic maximization problems, namely, the longest path and the maximum independent set, are studied. In [20,19], the Steiner forest problem and the classical Steiner tree problem are handled, respectively. An early survey about a priori optimization can be found in [2] while, a more recent one appears in [16].

As already mentioned, in probabilistic combinatorial optimization, the combinatorial problem to be solved, being subject to hazards or to inaccuracies, is not defined on a static and clearly specified instance, since the instance to be effectively optimized is not known with absolute certainty from the beginning. The goal here is to compute solutions that behave "well" for any subset of the initial data-set. In this sense, a priori probabilistic combinatorial optimization can be seen as a particular case of stochastic programming addressed for combinatorial optimization problems.

There are two major computational challenges associated with a probabilistic combinatorial optimization problem:

- obtain a polynomial time computable expression for the objective function $E(G, S, \mathtt{M})$ (let us note that this function carries, by (1), over 2^n additive terms; henceforth the complexity of its computation is not trivially polynomial);
- give a closed tight combinatorial characterization of the optimal anticipatory solution (this implies the derivation of a compact combinatorial characterization of the solution optimizing (1)).

Our goal in this paper is to go beyond study of probabilistic versions of particular combinatorial problems and to propose a structural way to handle this model. Notice that, for any problem Π, its probabilistic counterpart, PROBABILISTIC Π, contains Π as subproblem (just consider probability vector $(1, \ldots, 1)$ for Π). Hence, from a complexity point of view, PROBABILISTIC Π is at least as hard as Π, that is, if Π is **NP**-hard, then PROBABILISTIC Π is also **NP**-hard, while if Π is polynomial, no immediate conclusion can be derived for the complexity of PROBABILISTIC Π, until this latter problem is explicitly studied.

In what follows, we consider the following very simple quick modification strategy M. *Given an anticipatory solution S and a subgraph $G' = G[V']$, S' is the restriction of S in G'. If S' is feasible for G', then retain it. If not, quickly "patch" S' (a possible patching will be specified later) in order to get a feasible solution for G'.*

We handle three categories of combinatorial graph-problems exhausting a very large part of the most known ones. In Section 3., we study problems whose solutions are subsets of the input vertex-set verifying some specific property. In Section 4., we handle problems whose solutions are collections of subsets of the input vertex-set verifying some specified non-trivial hereditary property[1]. In Section 5., we deal with problems whose solutions are subsets of the input edge-set verifying some specific property and the restriction of any anticipatory solution to any subgraph of the input-graph is feasible. For any of the categories considered in Sections 3., 4. and 5., we give sufficient conditions under which functionals are analytically expressible and polynomially computable and anticipatory solutions are well-characterized. Let us note also that for any problem in these categories, restriction of any anticipatory solution to any realized subgraph produce feasible solutions for the subgraph at hand.

In Section 6., things become more complicated since the problems handled have as solutions connected subsets of the input edge-set that are either cycles, or paths, or trees. For this type of problems, the restriction of the anticipatory solution S to $G[V']$ is not feasible in general and some additional work (with low algorithmic complexity) is needed in order to render this set feasible. Informally, it could be the case that restriction of the anticipatory solution S to the present subgraph $G' = G[V']$ leaves S non-connected. So, in order to produce a feasible solution for G', one has to reconnect the connected components of S. As we will see, in this case, anticipatory solutions cannot be as well- and compactly characterized as those of Sections 3., 4. and 5.. However, we give sufficient conditions under which functionals for the probabilistic counterparts of the concerned problems are computable in polynomial time.

The structural results given in the paper immediately apply to several well-known problems, for instance, MIN VERTEX COVER, MAX INDEPENDENT SET, MIN COLORING, MAX CUT, MAX MATCHING, MIN TSP, etc., producing particular results interesting per se. Furthermore, the scope of our results is even larger as they capture problems even defined on set-systems like the MIN SET COVER. So, this work can provide a framework for a systematic classification of a great number of probabilistic derivatives of well-known combinatorial optimization problems.

In what follows, we deal with problems in **NPO**. Informally, this class contains optimization problems whose decision versions belong to **NP**. Given a combinatorial problem $\Pi \in$ **NPO**, we denote by PROBABILISTIC Π, its probabilistic counterpart defined as described previously and assume that the vertex-probabilities are independent.

[1] A property π is *hereditary* if, whenever is satisfied by a graph G, it is satisfied by any subgraph of G; a hereditary property π is non-trivial if it is true (satisfied) for infinitely many graphs and false for infinitely many graphs.

Let $\mathtt{A}$ be a polynomial time approximation algorithm for an **NP**-hard graph-problem Π, let $m(G,S)$ be the value of the solution S provided by $\mathtt{A}$ on an instance G of Π, and opt(G) be the value of the optimal solution for G (following our notation for PROBABILISTIC Π, $m(G,S) = E(G,S,\mathtt{M})$ and opt$(G) = E(G,S^*,\mathtt{M})$). The approximation ratio $\rho_{\mathtt{A}}(G)$ of the algorithm $\mathtt{A}$ on G is defined by $\rho_{\mathtt{A}}(G) = m(G,S)/\text{opt}(G)$. An approximation algorithm achieving ratio, at most, ρ on any instance G of Π will be called ρ-approximation algorithm. Since modification strategy $\mathtt{M}$ used in each section of the paper is unique and fixed, it will be omitted for simplicity from the mathematical expressions.

3. Solutions are subsets of the initial vertex-set

In this section, we deal with graph-problems whose solutions are subsets of the vertex-set of the input-graph. We further assume that given a solution S and a set $V' \subseteq V$, the restriction of S in V', i.e., the set $S' = S \cap V'$ is feasible for $G[V']$. The main result of this section is stated in Proposition 1.

Proposition 1 *Consider a graph-problem Π verifying the following assumptions: (i) an instance of Π is a vertex-weighted graph $G(V,E,\vec{w})$; (ii) solutions of Π are subsets of V; (iii) for any solution S and any subset $V' \subseteq V$, $S' = S \cap V'$ is feasible for $G' = G[V']$; (iv) the value of any solution $S \subseteq V$ is defined by: $m(G,S) = w(S) = \sum_{v_i \in S} w_i$, where w_i is the weight of $v_i \in V$. Then, the functional of* PROBABILISTIC Π *is expressed as: $E(G,S) = \sum_{v_i \in S} w_i p_i$ and can be computed in polynomial time. Furthermore, the complexity of* PROBABILISTIC Π *is the same as the one of* Π.

Proof. Fix a subset $V' \subseteq V$ and an anticipatory solution S for PROBABILISTIC Π on G. According to assumptions (iii) and (iv), S' is feasible for $G[V']$ and its value is given by: $m(G',S') = \sum_{v_i \in S} w_i 1_{\{v_i \in V'\}}$. Then, denoting by 1_F the indicator function of a fact F and using (1) we get:

$$\begin{aligned} E(G,S) &= \sum_{V' \subseteq V} m\left(G',S'\right) \Pr\left[V'\right] \\ &= \sum_{V' \subseteq V} \sum_{v_i \in S} w_i 1_{\{v_i \in V'\}} \Pr\left[V'\right] \\ &= \sum_{v_i \in S} w_i \sum_{V' \subseteq V} 1_{\{v_i \in V'\}} \Pr\left[V'\right] \end{aligned} \quad (2)$$

For any vertex $v_i \in V$, let $V_i = V \setminus \{v_i\}$ and $\mathcal{V}_i' = \{V' \subseteq V : V' = \{v_i\} \cup V'', V'' \subseteq V_i\}$. Using also the fact that presence-probabilities of the vertices of V are independent, we get:

$$\begin{aligned} \sum_{V' \subseteq V} 1_{\{v_i \in V'\}} \Pr\left[V'\right] &= \sum_{V' \in \mathcal{V}_i'} \Pr\left[V'\right] \\ &= \sum_{V'' \subseteq V_i} \Pr\left[\{v_i\} \cup V''\right] \\ &= \sum_{V'' \subseteq V_i} \Pr\left[v_i\right] \Pr\left[V''\right] \\ &= \Pr\left[v_i\right] \sum_{V'' \subseteq V_i} \Pr\left[V''\right] = p_i \end{aligned} \quad (3)$$

Combination of (2) and (3) immediately leads to the expression claimed for $E(G,S)$.

It is easy to see that this functional can be computed in time linear in n. Furthermore, computation of the optimal anticipatory solution for PROBABILISTIC Π in G, obviously amounts to the computation of the optimal weighted solution for Π in $G(V,E,\vec{w}')$, where, for any $v_i \in V$, $w_i' = w_i p_i$. Consequently, by this observation and by assumption (iv), Π and PROBABILISTIC Π have the same worst case complexity. ■

Although computation of the functional is, as we have mentioned, a priori exponential (since it carries over the 2^n possible subgraphs of G), assumptions (i) through (iv) in Proposition 1 allow polynomial computation of its value. This is due to the fact that, under these assumptions, given a subgraph G' induced by a subset $V' \subseteq V$, the value of the solution for G' is the sum of the weights of the vertices in $S \cap V'$. Furthermore, a vertex not in S will never make part of any solution in any sub-graph of G. Consequently, computation of the functional amounts to determining, for any G', which vertices make part of $S \cap V'$. This can be done by specifying, for any $v_i \in S$, all the subgraphs to which v_i belongs, and by performing a summation of the presence-probabilities of these subgraphs. This sum is equal to p_i (the probability of v_i). This simplification is the main reason that renders functional's computation polynomial, despite of the exponential number of terms in its generic expression.

Notice that Proposition 1 can also be used for getting generic approximation results for PROBABILISTIC Π. Indeed, since this problem is a particular weighted version of Π, one immediately concludes that *if Π is approximable within approximation ratio ρ, so is* PROBABILISTIC Π.

Corollary 1 *Under the hypotheses of Proposition 1, whenever Π and* PROBABILISTIC Π *are **NP**-hard, they* *are equi-approximable.*

Proposition 1 has also the following immediate corollary dealing with the case of probabilistic versions of unweighted combinatorial optimization problems.

Corollary 2 *Consider a problem* Π *verifying assumptions (i) to (iv) of Proposition 1 with* $\vec{w} = \vec{1}$*. Then, the functional of* PROBABILISTIC Π*, is expressed as:* $E(G, S) = \sum_{v_i \in S} p_i$ *and can be computed in polynomial time. Furthermore,* PROBABILISTIC Π *is equivalent to a weighted version of* Π *where vertex-weights are the vertex-probabilities.*

Corollary 2 is weaker than Proposition 1 since it simply establishes a kind of (obvious) reduction from Π to PROBABILISTIC Π stating that whenever Π is **NP**-hard, so is PROBABILISTIC Π. However, if Π is polynomial, the status of PROBABILISTIC Π remains unclear by Corollary 2.

Proposition 1 can be applied to a broad class of problems that fit its four conditions, as PROBABILISTIC MAX INDEPENDENT SET ([14]), PROBABILISTIC MIN VERTEX COVERING ([15]), etc. We describe in what follows two further applications, namely, PROBABILISTIC MAX INDUCED SUBGRAPH WITH PROPERTY π and PROBABILISTIC MIN FEEDBACK VERTEX-SET.

3.1. PROBABILISTIC MAX INDUCED SUBGRAPH WITH PROPERTY π

Consider a graph $G(V, E)$ and a non-trivial hereditary property. A feasible solution for MAX INDUCED SUBGRAPH WITH PROPERTY π is a subset $V' \subseteq V$ such that, the subgraph $G[V']$ of G induced by V' satisfies π. The objective is to determine such a set V' of maximum-size. Note that, "*independent set*", "*clique*", "*planar graph*" are hereditary properties. In the weighted version of the problem (i.e., the one where positive weights are associated with the vertices of G), called MAX WEIGHTED INDUCED SUBGRAPH WITH PROPERTY π, we search for maximizing the total weight of V'.

Given a solution S for MAX WEIGHTED INDUCED SUBGRAPH WITH PROPERTY π and an induced subgraph $G[V']$ of the input graph $G(V, E)$, the set $S \cap V'$ is a feasible solution for $G[V']$, since, by the definition of π, if a subset $S \subseteq V$ induces a subgraph verifying it, then any subset of S also induces a subgraph verifying π. Henceforth, MAX WEIGHTED INDUCED SUBGRAPH WITH PROPERTY π fits the conditions of Proposition 1.

3.2. PROBABILISTIC MIN FEEDBACK VERTEX-SET

Given an oriented graph $G(V, A)$, a *feedback vertex-set* is a subset $V' \subseteq V$ such that V' contains at least a vertex of any directed cycle of G. In MIN FEEDBACK VERTEX-SET, the objective is to determine a feedback vertex-set of minimum size.

Remark that, absence of a vertex v from a feedback vertex-set V', breaks any cycle containing this vertex. If v makes part of an anticipatory solution S then, since no such cycle that contained v exists in G', feasibility of the solution $S \cap V'$ does not suffer from the absence of v. So, Corollary 2 applies for this problem.

Note that the weighted version of this problem can be tackled in a similar way.

4. Solutions are collections of subsets of the initial vertex-set

We now handle problems the feasible solutions of which are collections of subsets of the initial vertex-set. Consider a graph $G(V, E)$ and a combinatorial optimization graph-problem Π whose solutions are collections of subsets of V verifying some specified non-trivial hereditary property. The following theorem characterizes functionals and optimal anticipatory solutions for such problems.

Proposition 2 *Consider a graph-problem* Π *verifying the following assumptions: (i) an instance of* Π *is a graph* $G(V, E)$*; (ii) a solution of* Π *on* G *is a collection* $S = (V_1, \ldots, V_k)$ *of subsets of* V *each of them satisfying some specified non-trivial hereditary property; (iii) for any solution* S *and any subset* $V' \subseteq V$*, the restriction* S' *of* S *in* V'*, i.e.,* $S' = (V_1 \cap V', \ldots, V_k \cap V')$*, is feasible for* $G' = G[V']$*; (iv) the value of any solution* $S \subseteq V$ *of* Π *is defined by:* $m(G, S) = |S| = k$*. Then,* $E(G, S) = \sum_{j=1}^{k} (1 - \prod_{v_i \in V_j} (1 - p_i))$ *and can be computed in polynomial time.* PROBABILISTIC Π *amounts to a particular weighted version of* Π*, where the weight of any vertex* $v_i \in V$ *is* $1 - p_i$*, the weight* $w(V_j)$ *of a subset* $V_j \subseteq V$ *is defined by* $w(V_j) = 1 - \prod_{v_i \in V_j} (1 - p_i)$ *and the objective function to be optimized is equal to* $\sum_{V_j \in S} w(V_j)$*.*

Proof. Consider an anticipatory solution $S = (V_1, V_2, \ldots, V_k)$ and a subgraph $G' = G[V']$ of G. Denote by $k' = m(G', S')$, the value of the solution obtained on G' as described in assumption (iii). Then, $E(G, S) = \sum_{V' \subseteq V} \Pr[V'] k'$.

Consider the facts: F_j: $V_j \cap V' \neq \emptyset$ and $\bar{F}_j$: $V_j \cap V' = \emptyset$. Then, k' can be written as $k' = \sum_{j=1}^{k} 1_{F_j} =$

$\sum_{j=1}^{k}(1 - 1_{\bar{F}_j})$ and $E(G, S)$ becomes:

$$E(G,S) = \sum_{V' \subseteq V} \Pr[V'] \left(\sum_{j=1}^{k} \left(1 - 1_{\bar{F}_j} \right) \right)$$
$$= \sum_{V' \subseteq V} \Pr[V'] \sum_{j=1}^{k} 1 -$$
$$\sum_{V' \subseteq V} \Pr[V'] \sum_{j=1}^{k} 1_{V_j \cap V' = \emptyset}$$
$$= \sum_{j=1}^{k} \sum_{V' \subseteq V} \Pr[V'] - \sum_{j=1}^{k} \sum_{V' \subseteq V} \Pr[V'] \, 1_{V_j \cap V' = \emptyset}$$
$$= k - \sum_{j=1}^{k} \prod_{v_i \in V_j} (1 - p_i)$$
$$= \sum_{j=1}^{k} \left(1 - \prod_{v_i \in V_j} (1 - p_i) \right) \qquad (4)$$

It is easy to see that computation of $E(G, S)$ can be performed in at most $O(n)$ steps; consequently, PROBABILISTIC Π is in **NPO**. Furthermore, by (4), the characterization of the feasible solutions for PROBABILISTIC Π claimed in the statement of the proposition is immediate. ■

Central role for yielding result of Proposition 2 plays the fact that the property satisfied by the sets of the collection S is hereditary. This allows to the non-empty sets of the restriction of S to V' to be a feasible solution for $G[V']$ and, consequently, to express $E(G, S)$ as in (4), using the facts F_j and $\bar{F}_j$.

Assume that $p_i = 1$, for any $v_i \in V$. Then, by (4), $E(G, S) = k$ and PROBABILISTIC Π coincides in this case with Π.

Corollary 3 *If Π is **NP**-hard, then* PROBABILISTIC Π *is also **NP**-hard.*

As for Corollary 2, Corollary 3 settles complexity only for the case where Π is **NP**-hard, leaving unclear the status of PROBABILISTIC Π when $\Pi \in$ **P**.

Proposition 2 also captures numerous combinatorial optimization problems, as PROBABILISTIC MIN COLORING ([17]), PROBABILISTIC MIN PARTITION INTO CLIQUES, etc. In what follows, we describe two further applications, namely, PROBABILISTIC MIN COMPLETE BIPARTITE SUBGRAPH COVER and PROBABILISTIC MIN CUT COVER. Then, we show that Proposition 2 can go beyond graphs by giving a formulation of MIN SET COVER as a graph-problem and proving that, according to this formulation, PROBABILISTIC MIN SET COVER also fits conditions of Proposition 2.

4.1. PROBABILISTIC MIN COMPLETE BIPARTITE SUBGRAPH COVER

Given a graph $G(V, E)$, a solution of MIN COMPLETE BIPARTITE SUBGRAPH COVER is a collection $\mathcal{C} = (V_1, V_2, \ldots, V_k)$ of subsets of V such that the subgraph induced by any of the V_i's, $i = 1, \ldots, k$, is a complete bipartite graph and for any edge $(u, v) \in E$ there exists a V_i containing both u and v. The objective here is to minimize the size $|\mathcal{C}|$ of $\mathcal{C}$.

Remark first that the property "complete bipartite graph" is hereditary. Consider a solution $\mathcal{C} = (V_1, \ldots, V_k)$ of MIN COMPLETE BIPARTITE SUBGRAPH COVER and a subset $V' \subseteq V$. The set $\mathcal{C}' = (V_1 \cap V', \ldots, V_k \cap V')$, is feasible for $G' = G[V']$. Indeed, if a vertex v disappears from some set V_i of an anticipatory solution $\mathcal{C}$, the surviving set $V_i \setminus \{v\}$ always induces a complete bipartite graph. Furthermore, except for the edges that have been disappeared (the ones incident to v) any other edge remain covered by the surviving sets of $\mathcal{C}$.

So, PROBABILISTIC MIN COMPLETE BIPARTITE SUBGRAPH COVER meets the conditions of Proposition 2.

4.2. PROBABILISTIC MIN CUT COVER

Given a graph $G(V, E)$, a feasible solution for MIN CUT COVER is a collection $(V_1, \ldots, V_k)$ of V such that any V_i, $i = 1, \ldots, k$ is a cut, i.e., for any $(u, v) \in E$, there exists a V_i such that either $u \in V_i$ and $v \notin V_i$, or $u \notin V_i$ and $v \in V_i$. The objective is to minimize the size k of the collection.

Consider a solution $S = (V_1, \ldots, V_k)$ for MIN CUT COVER. If a vertex $v \in V$ is absent, then any edge incident to v is also absent. So, the edges of the final graph $G'(V', E')$, remain feasibly covered by the restriction of S to V'. Hence, $S' = (V_1 \cap V', \ldots, V_k \cap V')$ is feasible for MIN CUT COVER, that meets the conditions of Proposition 2, since property "cut" is hereditary.

4.3. PROBABILISTIC MIN SET COVER

Given a collection $\mathcal{S} = \{S_1, \ldots S_m\}$ of subsets of a ground set $C = \{c_1, \ldots, c_n\}$ (it is assumed that $\cup_{S_i \in \mathcal{S}} S_i = C$), MIN SET COVERING consists of determining a minimum-size set cover of C, i.e., a minimum-size sub-collection $\mathcal{S}'$ of $\mathcal{S}$ such that $\cup_{S_i \in \mathcal{S}'} S_i = C$.

Starting from an instance $(\mathcal{S}, C)$ of MIN SET COVER, one can construct an edge-colored multi-graph $G_C(V_C, E_C, \vec{\ell}_{\mathcal{S}})$ as follows: for any $c_i \in C$, add a vertex $v_i \in V_C$; for any pair c_i, c_j of elements in C, add a new edge (v_i, v_j) colored with S_k only if $S_k \supseteq \{c_i, c_j\}$.

In the so-constructed graph G_C a set $S_i = \{c_{i_1}, \dots c_{i_k}\} \in \mathcal{S}$ becomes a clique on vertices $v_{i_1}, \dots, v_{i_k} \in V_C$ all the edges of which are colored with the same color S_i; we will call such a clique a unicolored clique. Under this formulation, MIN SET COVER can be viewed as a particular clique-covering problem where the objective is to determine a minimum size cover of V_C by unicolored cliques.

Consider a set cover $\mathcal{S}'$ for the initial instance $(\mathcal{S}, C)$ and a sub-instance I' of $(\mathcal{S}, C)$ consisting of some elements of C and of the subsets of $\mathcal{S}$ including these elements. These objects correspond, in G_C, to a vertex-covering by unicolored cliques and the subgraph G'_C of G_C defined with respect to I'. Restriction of $\mathcal{S}'$ in I' can be viewed, with respect to G_C, as restriction of the initial vertex-covering by unicolored cliques to the vertices of G'_C. Observe finally that "unicolored clique" is a hereditary property. So, under this formulation, PROBABILISTIC MIN SET COVER perfectly fits conditions of Proposition 2.

According to the formulation used for MIN SET COVER, given an instance $(\mathcal{S}, C)$ with element-probabilities p_i, for any $c_i \in C$, and a feasible solution $\mathcal{S}'$ of $(\mathcal{S}, C)$, then, $E((\mathcal{S}, C), \mathcal{S}') = \sum_{S_i \in \mathcal{S}'} (1 - \prod_{c_j \in S_i} (1 - p_j))$ and can be computed in polynomial time. The probabilistic version of MIN SET COVER amounts to a particular weighted version of the initial problem where each set $S_i = \{c_{i_1}, \dots, c_{i_k}\}$ in $\mathcal{S}$ is weighted by $1 - \prod_{j=1}^{k} (1 - p_{i_j})$.

Hence, PROBABILISTIC MIN SET COVER is indeed a simple weighted version of MIN SET COVER, where one has to determine a set cover minimizing its total weight. In this sense, the problem dealt seems to be simpler than the majority of the problems captured by Proposition 2 as, for instance, MIN COLORING. This is due to the fact that, dealing with MIN SET COVER, there is a polynomial number of unicolored cliques in G_C (the sets of $\mathcal{S}$) candidate to be part of any solution, while, for MIN COLORING the number of the independent sets that may be part of a solution is exponential.

4.4. *A generic approximation result for the problems fitting conditions of Proposition 2*

This section extends an approximation result of [17] for PROBABILISTIC MIN COLORING, in order to capture the whole of problems meeting the conditions of Proposition 2.

Consider such an **NPO** problem Π, an instance $G(V, E)$ of Π, set $n = |V|$ and consider a solution $S = (V_1, \dots, V_k)$ of Π on G (recall that $V_1, \dots, V_k$ are assumed mutually disjoint). Denote by $p_{\min}$ and $p_{\max}$ the minimum and maximum vertex-probabilities, respectively. Then, the following bounds hold for $E(G, S)$:

$$\max\left\{\sum_{i=1}^{n} p_i - \sum_{i=1}^{n} \sum_{j=i+1}^{n} p_i p_j, k p_{\min}\right\} \leqslant E(G, S) \leqslant \min\left\{\sum_{i=1}^{n} p_i \leqslant n p_{\max}, k\right\} \quad (5)$$

Observe first that the rightmost upper bound for $E(G, S)$ in (5) is immediately derived from the expression for $E(G, S)$ in the statement of Proposition 2.

We now prove the leftmost upper bound and the lower bounds of (5). We first produce a framing for the term $1 - \prod_{v_i \in V_j} (1 - p_i)$. For simplicity, assume $|V_j| = \ell$ and arbitrarily denote vertices in V_j by $v_1, \dots, v_\ell$. By induction on ℓ, we show that:

$$\sum_{i=1}^{\ell} p_i - \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} p_i p_j \leqslant 1 - \prod_{i=1}^{\ell} (1 - p_i) \leqslant \sum_{i=1}^{\ell} p_i \quad (6)$$

For the left-hand side of (6), observe first that it is true for $\ell = 1$ and suppose it true for $\ell = \kappa$, i.e., $\sum_{i=1}^{\kappa} p_i - \sum_{i=1}^{\kappa} \sum_{j=i+1}^{\kappa} p_i p_j \leqslant 1 - \prod_{i=1}^{\kappa} (1 - p_i)$, or:

$$\prod_{i=1}^{\kappa} (1 - p_i) \leqslant 1 - \sum_{i=1}^{\kappa} p_i + \sum_{i=1}^{\kappa} \sum_{j=i+1}^{\kappa} p_i p_j \quad (7)$$

Suppose now that $\ell = \kappa + 1$ and multiply both terms of (7) by $(1 - p_{\kappa+1})$; then:

$$\prod_{i=1}^{\kappa+1}(1-p_i) \leqslant \left(1-\sum_{i=1}^{\kappa} p_i+\sum_{i=1}^{\kappa}\sum_{j=i+1}^{\kappa} p_ip_j\right)(1-p_{\kappa+1})$$
$$= 1 - \sum_{i=1}^{\kappa} p_i + \sum_{i=1}^{\kappa}\sum_{j=i+1}^{\kappa} p_ip_j - p_{\kappa+1} + p_{\kappa+1}\sum_{i=1}^{\kappa} p_i - p_{\kappa+1}\sum_{i=1}^{\kappa}\sum_{j=i+1}^{\kappa} p_ip_j$$
$$= 1 - \sum_{i=1}^{\kappa+1} p_i + \sum_{i=1}^{\kappa+1}\sum_{j=i+1}^{\kappa+1} p_ip_j - p_{\kappa+1}\sum_{i=1}^{\kappa}\sum_{j=i+1}^{\kappa} p_ip_j$$
$$\leqslant 1 - \sum_{i=1}^{\kappa+1} p_i + \sum_{i=1}^{\kappa+1}\sum_{j=i+1}^{\kappa+1} p_ip_j$$

that proves the left-hand side inequality in (6).

For the right-hand side of (6), we show by induction on ℓ that $\prod_{i=1}^{\ell}(1 - p_i) \geqslant 1 - \sum_{i=1}^{\ell} p_i$. This is clearly true for $\ell = 1$. Suppose it also true for any $\ell \leqslant \kappa$, i.e., $\prod_{i=1}^{\kappa}(1-p_i) \geqslant 1-\sum_{i=1}^{\kappa} p_i$. Then, by multiplying both members of this inequality by $(1 - p_{\kappa+1})$, we get that the product obtained is equal to $1 - p_{\kappa+1} - \sum_{i=1}^{\kappa} p_i + p_{\kappa+1}\sum_{i=1}^{\kappa} p_i \geqslant 1 - \sum_{i=1}^{\kappa+1} p_i$, q.e.d.

Remark 1 *Let us note that (6) is a special case of the following well-known result of the inclusion-exclusion principle: if* $\Pr(A_i) = p_i$, *then* $\Pr(\cap_i \bar{A}_i) = S_0 - S_1 + S_2 - S_3 + \ldots$ *where:*

$$S_k = \sum_{i_1<i_2<\ldots i_k} Pr\left(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}\right)$$

and

$$S_0-S_1+S_2-\ldots-S_{2k-1} \leq Pr\left(\cap \bar{A}_i\right) \leq S_0-S_1+\ldots+S_{2k}$$

Inequality (6) is the case where all A_i *are independent, and* $k = 1$.

Taking the sums of the members of (6) for $m = 1$ to k, the right-hand side inequality immediately gives $E(G, S) \leqslant \sum_{i=1}^{n} p_i$.

We now prove that $E(G, S) \geqslant \sum_{i=1}^{n} p_i - \sum_{i=1}^{n}\sum_{j=i+1}^{n} p_ip_j$ (the leftmost lower bound claimed in (5)). From the left-hand side of (6), we get:

$$\sum_{m=1}^{k}\left(\sum_{i=1}^{\ell} p_i - \sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell} p_ip_j\right) = \sum_{i=1}^{n} p_i - \sum_{m=1}^{k}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell} p_ip_j \geqslant \sum_{i=1}^{n} p_i - \sum_{i=1}^{n}\sum_{j=i+1}^{n} p_ip_j \quad (8)$$

Observe that, from the first inequality of (6), we have:

$$\sum_{m=1}^{k}\left(\sum_{i=1}^{\ell} p_i-\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell} p_ip_j\right) \leqslant \sum_{m=1}^{k}\left(1-\prod_{i=1}^{\ell}(1-p_i)\right) \quad (9)$$

The righthand side of (9) is exactly $E(G, S)$. Putting this together with (8), the leftmost lower bound for $E(G, S)$ in (5) is proved.

Finally, in order to derive the rightmost lower bound in (5), observe that $\prod_{v_i \in S_j}(1 - p_i) \leqslant (1 - p_{\min})^{|S_j|} \leqslant 1 - p_{\min}$, i.e., $1 - \prod_{v_i \in S_j}(1 - p_i) \geqslant p_{\min}$. Summing for $j = 1$ to k, we get the bound claimed.

We are ready now to study an approximation algorithm for the whole class of problems meeting Proposition 2. Fix a vertex-probability p', assume that there exists a ρ-approximation polynomial time algorithm A for Π, and run the following algorithm, called RA for PROBABILISTIC Π:

(1) partition the vertices of G into three subsets: the first, V_1 including the vertices with probabilities at most $1/n$, the second, V_2, including the vertices with probabilities in the interval $[1/n, p']$ and the third, V_3, including the vertices with probabilities greater than p';
(2) feasibly solve Π in $G[V_1]$ and $G[V_2]$ separately;
(3) run A in $G[V_3]$;
(4) take the union of the solutions computed in steps 2 and 3 as solution for G.

Theorem 1 *If A achieves approximation ratio* ρ *for* Π, *then RA approximately solves in polynomial time* PROBABILISTIC Π *within ratio* $O(\sqrt{\rho n})$.

Proof. Denote by $S^* = (V_1^*, \ldots, V_{k^*}^*)$ an optimal anticipatory solution, by $S = (\hat{V}_1, \ldots, \hat{V}_k)$ the approximate anticipatory solution computed in step 4 and, respectively, by $S_i^* = (V_{1,i}^*, \ldots, V_{|S_i^*|,i}^*)$ and $S_i = (\hat{V}_{1,i}, \ldots, \hat{V}_{|S_i|,i})$, the optimal and approximate solutions in $G[V_i]$, $i = 1, 2, 3$. Denote by $S^*[V_1]$, $S^*[V_2]$ and $S^*[V_3]$ the restrictions of S^* in $G[V_1]$, $G[V_2]$ and $G[V_3]$, respectively. Denote finally by n_i, the orders of $G[V_i]$, for $i = 1, 2, 3$, respectively. The proof is based upon the following claims.

(1) Any feasible polynomial time approximation algorithm for PROBABILISTIC Π achieves in the graph $G[V_1]$ approximation ratio bounded above by 2.
(2) Any feasible polynomial time approximation algorithm for PROBABILISTIC Π achieves in the graph $G[V_2]$ approximation ratio bounded above by $O(np')$.
(3) Assuming that A achieves approximation ratio ρ for Π, when running in $G[V_3]$ it achieves approximation ratio bounded above by ρ/p' for PROBABILISTIC Π.

For Claim 1, using (5) for S_1 and S_1^*, we get: $E(G[V_1], S_1) \leqslant \sum_{i=1}^{n_1} p_i$ and $E(G[V_1], S_1^*) \geqslant \sum_{i=1}^{n_1} p_i - \sum_{i=1}^{n_1}\sum_{j=i+1}^{n_1} p_i p_j$. Combining them, we derive:

$$\frac{E\left(G\left[V_1\right], S_1^*\right)}{E\left(G\left[V_1\right], S_1\right)} \geqslant 1 - \frac{\sum_{i=1}^{n_1}\sum_{j=i+1}^{n_1} p_i p_j}{\sum_{i=1}^{n_1} p_i} = 1 - \frac{\left(\sum_{i=1}^{n_1} p_i\right)^2 - \sum_{i=1}^{n_1} p_i^2}{2\sum_{i=1}^{n_1} p_i} \geqslant 1 - \frac{\sum_{i=1}^{n_1} p_i}{2} + \frac{\sum_{i=1}^{n_1} p_i^2}{2\sum_{i=1}^{n_1} p_i} \geqslant 1 - \frac{\sum_{i=1}^{n_1} p_i}{2} \tag{10}$$

Since p_i's are smaller than $1/n$ and $n_1 \leqslant n$, the right-hand side of (10) is at least as large as $1/2$. Hence, every algorithm for Π in $G[V_1]$ achieves ratio $E(G[V_1], S_1)/E(G[V_1], S_1^*) \leqslant 2$ for PROBABILISTIC Π, and the proof of Claim 1 is complete.

We now prove Claim 2. Here, for any v_i, $p_i \geqslant 1/n$. Consequently, $1 - \prod_{v_i \in V_{j,2}^*}(1 - p_i) \geqslant 1 - (1 - (1/n))^{|V_{j,2}^*|} \geqslant (|V_{j,2}^*|/n) - (|V_{j,2}^*|(|V_{j,2}^*| - 1)/2n^2)$, where the last inequality is an easy application of the left-hand side of (6) with $p_i = 1/n$ for any vertex v_i. Furthermore:

$$\frac{\left|V_{j,2}^*\right|}{n} - \frac{\left|V_{j,2}^*\right|\left(\left|V_{j,2}^*\right| - 1\right)}{2n^2} = \frac{\left|V_{j,2}^*\right|}{n}\left(1 - \frac{\left|V_{j,2}^*\right| - 1}{2n}\right) \geqslant \frac{\left|V_{j,2}^*\right|}{n} \times \frac{n+1}{2n} \geqslant \frac{\left|V_{j,2}^*\right|}{2n} \tag{11}$$

Summing inequality (11) for $j = 1, \ldots, |S_2^*|$, we get $E(G[V_2], S_2^*) \geqslant n_2/2n$, where n_2 is the order of $G[V_2]$. On the other hand, using the leftmost upper bound in (5), we get $E(G[V_2], S_2) \leqslant n_2 p'$. The bounds for $E(G[V_2], S_2^*)$ and $E(G[V_2], S_2)$ immediately derive approximation ratio at most $2np' = O(np')$ for every algorithm solving PROBABILISTIC Π in $G[V_2]$ and the proof of Claim 2 is complete.

We now turn to Claim 3. Using the rightmost lower bound of (5), $E(G[V_3], S_3^*) \geqslant |S_3^*|p'$. On the other hand, by the rightmost upper bound of (5), $E(G[V_3], S_3) \leqslant |S_3|$. So, assuming that A achieves ratio ρ for Π, step 3 achieves ratio $(|S_3|/|S_3^*|)p'$ for $G[V_3]$, that turns out to a ratio bounded above by ρ/p' for PROBABILISTIC Π, completing so the proof of Claim 3.

We prove that, for any $k \in \{1, 2, 3\}$: $E(G, S^*) \geqslant E(G[V_k], S^*[V_k]) \geqslant E(G[V_k], S_k^*)$. Remark that $S^*[V_k]$ is a particular feasible solution for $G[V_k]$; hence: $E(G[V_k], S^*[V_k]) \geqslant E(G[V_k], S_k^*)$. In order to prove the first inequality, fix a k and consider a component, say V_j^* of S^*. Then, the contribution of V_j^* in $S^*[V_k]$ is: $1 - \prod_{v_i \in V_j^* \cap V_k}(1 - p_i) \leqslant 1 - \prod_{v_i \in V_j^*}(1 - p_i)$, that is its contribution in S^*. Iterating this argument for all the elements in $S^*[V_k]$, the claim follows.

Algorithm RA solves separately each $G[V_k]$, $k \in \{1, 2, 3\}$ and returns as solution S the union of the solutions computed in the three induced subgraphs. Hence, $E(G, S) = E(G[V_1], S_1) + E(G[V_2], S_2) + E(G[V_3], S_3)$. Furthermore, $E(G, S^*)$ is at least as large as any of $E(G[V_k], S_k^*)$, $k \in \{1, 2, 3\}$. So, the ratio of the algorithm in G is at most the sum of the ratios proved by Claims 1, 2 and 3, i.e., at most $O(2 + np' + (\rho/p'))$.

Note that the ratio claimed in Claim 2 is increasing with p', while that of Claim 3 is decreasing with p'. Equality of expressions np' and ρ/p' holds for $p' = \sqrt{\rho/n}$. In this case the value of the ratio obtained is $O(\sqrt{\rho n})$, and the proof of the theorem is now completed. ■

5. Solutions are subsets of the initial edge-set

We now handle problems for which solutions are sets of edges. Notice that whenever a vertex is absent from some subset $V' \subseteq V$, the edges incident to it are also absent from $G[V']$.

Proposition 3 *Consider a graph-problem Π verifying the following assumptions: (i) an instance of Π is an edge- (or arc-) valued graph $G(V, E, \vec{\ell})$;*

(ii) any solution of Π *on any instance* G *is a subset of* E*; (iii) for any solution* S *and any subset* $V' \subseteq V$*, denoting by* $G'(V', E')$ *the subgraph of* G *induced by* V'*, the set* $S \cap E'$ *is feasible; (iv) the value of any solution* $S \subseteq E$ *of* Π *is defined by:* $m(G,S) = w(S) = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)$*, where* $\ell(v_i,v_j)$ *is the valuation of* $(v_i,v_j) \in E$*. Then,* $E(G,S) = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)p_ip_j$ *and can be computed in polynomial time. Furthermore, dealing with their respective computational complexities,* PROBABILISTIC Π *and* Π *are equivalent.*

Proof. Set $S' = S \cap E'$. By the assumptions of the proposition, S' is feasible for G'. Furthermore, $m(G',S') = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j) 1_{\{(v_i,v_j)\in E'\}}$. Then, using (1):

$$\begin{aligned} E(G,S) &= \sum_{V'\subseteq V} m\left(G',S'\right) \Pr\left[V'\right] \\ &= \sum_{V'\subseteq V} \sum_{(v_i,v_j)\in S} \ell\left(v_i,v_j\right) 1_{\{(v_i,v_j)\in E'\}} \Pr\left[V'\right] \\ &= \sum_{(v_i,v_j)\in S} \ell\left(v_i,v_j\right) \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in E'\}} \Pr\left[V'\right] \end{aligned} \tag{12}$$

Every $(v_i,v_j) \in E$ belongs to $G' = G[V']$, if and only if both of its endpoints belong to V'. Let $V_{ij} = V \setminus \{v_i,v_j\}$ and $\mathcal{V}'_{ij} = \{V' \subseteq V : V' = \{v_i\} \cup \{v_j\} \cup V'', V'' \subseteq V_{ij}\}$ be the set of all the subsets of V containing both v_i and v_j. Using also the fact that presence-probabilities of the vertices of V are independent, we get:

$$\begin{aligned} \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in E'\}} \Pr\left[V'\right] &= \sum_{V'\in\mathcal{V}'_{ij}} \Pr\left[V'\right] \\ &= \sum_{V''\subseteq V_{ij}} \Pr\left[\{v_i\}\cup\{v_j\}\cup V''\right] \\ &= \sum_{V''\subseteq V_{ij}} p_ip_j \Pr\left[V''\right] \\ &= p_ip_j \sum_{V''\subseteq V_{ij}} \Pr\left[V''\right] = p_ip_j \end{aligned} \tag{13}$$

Combination of (12) and (13) immediately leads to the expression claimed for the functional.

It is easy to see that this functional can be computed in time quadratic with n. Furthermore, computation of an optimal anticipatory solution for PROBABILISTIC Π in G obviously amounts to computation of an optimal solution for Π in an edge- (or arc-) valued graph $G(V,E,\vec{\ell'})$ where, for any $(v_i,v_j) \in E$, $\ell'(v_i,v_j) = \ell(v_i,v_j)p_ip_j$. Consequently, Π and PROBABILISTIC Π have the same complexity. ■

The reasons for which the functional derived in Proposition 3 becomes polynomial are quite analogous to those of Proposition 1. Since an edge that does not belong to the anticipatory solution S will never be part of $S \cap E'$ in any subgraph $G'(V',E')$ of G, the computation of the functional amounts to the quantification, for any G', of the average cardinality of the set $S \cap E'$. For this, it suffices to first determine, for any edge $e \in S$, all the subgraphs containing e and next to sum the probabilities of these subgraphs. This sum equals the product of the probabilities of the endpoints of e.

Let us note that, as in Section 3., Proposition 3 can be used for getting generic approximation results for PROBABILISTIC Π. Since this problem is a particular weighted version of Π (recall that Π is also a weighted problem), one immediately concludes that *if* Π *is approximable within approximation ratio* ρ*, so is* PROBABILISTIC Π.

Corollary 4 *Under the hypotheses of Proposition 3, whenever* Π *and* PROBABILISTIC Π *are* ***NP****-hard, they are equi-approximable.*

Corollary 5 *Consider a problem* Π *verifying assumptions (i) through (iv) of Proposition 3 with* $\vec{\ell} = \vec{1}$*. Then,* $E(G,S) = \sum_{(v_i,v_j)\in S} p_ip_j$ *and can be computed in polynomial time.* PROBABILISTIC Π *is equivalent to an edge- (or arc-) valued version of* Π *where the value of an edge is the product of the probabilities of its endpoints.*

As for Corollary 2, Corollary 5 does not conclude something definite for the complexity of PROBABILISTIC Π when Π is polynomial.

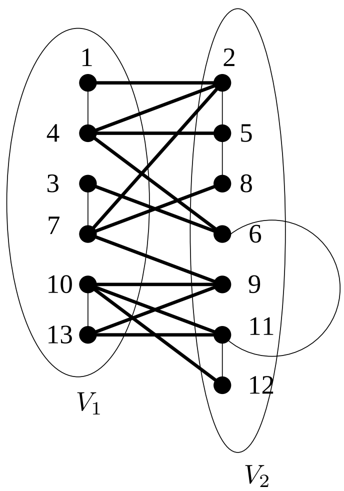
5.1. PROBABILISTIC MAX MATCHING

In MAX MATCHING, the objective is, given a graph $G(V,E)$ to determine a maximum-size matching, i.e., a maximum-size subset of E such that its edges are pairwise disjoint (they have no common endpoint).

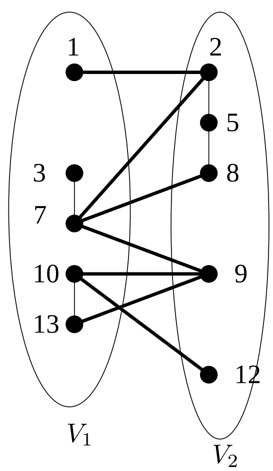
Clearly, MAX MATCHING in both edge-valued and non-valued graphs, fits conditions of Proposition 3 and Corollary 5, respectively. Moreover, since MAX WEIGHTED MATCHING is polynomial, both PROBABILISTIC MAX WEIGHTED MATCHING and PROBABILISTIC MAX MATCHING are also polynomial.

5.2. PROBABILISTIC MAX CUT

Consider a graph $G(V, E)$. In MAX CUT (resp. MAX WEIGHTED CUT) we wish to determine a maximum cardinality (resp., maximum weight) cut, i.e., to partition V into two subsets V_1 and V_2 such that a maximum number of edges (resp., maximum-weight set of edges) have one of their endpoints in V_1 and the other one in V_2.



(a) A graph G with a cut S (thick edges)



(b) Some "surviving" subgraph and the "surviving" solution

Fig. 1. An example for PROBABILISTIC MAX CUT.

We can represent an anticipatory cut S as a set of edges in such a way that whenever $(v_i, v_j) \in S$, $v_i \in V_1$ and $v_j \in V_2$. For example, in Figure 1(a), where for simplicity values of edges are not mentioned, the cut partitions V in subsets $V_1 = \{1, 3, 4, 7, 10, 13\}$ and $V_2 = \{2, 5, 6, 8, 9, 11, 12\}$; the anticipatory cut S (thick edges) is then $S = \{(1, 2), (3, 6), (4, 2), (4, 5), (4, 6), \ldots, (13, 11)\}$ (edges are ordered in lexicographic order). In Figure 1(b), we present graph's and cut's states assuming that vertices 4, 6 and 11 are absent. The solution S' considered misses in all edges of S having at least one endpoint in $\{4, 6, 11\}$ but it obviously remains a feasible cut for the surviving graph.

Hence, both weighted and cardinality PROBABILISTIC MAX CUT meet the conditions of Proposition 3 and Corollary 5, respectively. Consequently, MAX CUT being **NP**-hard, PROBABILISTIC MAX WEIGHTED CUT and PROBABILISTIC MAX CUT are also **NP**-hard.

6. When things become complicated: solutions are trees, or cordless cycles

In this section we handle edge-weighted graph-problems where a feasible solution is either a path, or a tree, or a cordless cycle. It is easy to see that, given such a solution S and a set $V' \subseteq V$ inducing a subgraph $G[V'] = G'(V', E')$ of G, the set $S \cap E'$ may be not feasible for G'.

Consider a problem Π where a feasible solution is a path, or a tree, or a cordless cycle denoted by S. Consider that the vertices in S are ordered in some appropriate order. Assume that $S \cap E'$ is a set of $k = k(G')$ (in other words, k depends on the present graph G') connected subsets $C_1, C_2, \ldots, C_k$ of S but that $S'' = \cup_{i=1}^{k} C_i$ is not connected (i.e., S'' does not constitute a feasible solution for Π). The vertices of each $C_1, C_2, \ldots, C_k$ are ordered consistently with the chosen order of S.

We consider a kind of "completion" of S'' by additional edges linking, for $i = 1, \ldots, k-1$, the last vertex (in the ordering considered for S) of C_i with the first vertex of C_{i+1}. In other words, given S (representing a connected set of edges) and V', and assuming that vertices of S are ordered following some appropriate order, we apply the following algorithm, denoted by A in the sequel (recall that V' is ordered following the order considered for S):

1. compute $S \cap E'$; let $C_1, C_2, \ldots, C_k$ be the resulting connected components of $S \cap E'$;
2. for $i = 1, \ldots, k-1$, use an edge to link the last vertex v_p of C_i to the first vertex v_q of C_{i+1} (where p and q are indexes of vertices according to the chosen ordering for S), if $p < q$ in the order consid-

ered for S;

(3) output the obtained solution and denote it by S'.

Obviously, in order that step 2 of A is able to link components C_i and C_{i+1}, an edge must exist between the vertices implied; otherwise, A is definitely unfeasible. So, in order to assure feasibility, we make, for the rest of the section, the basic assumption that the input graph for the problems handled is complete.

In what follows, we denote by $V[S']$ the set of vertices in S' and set $G''(V[S'], E'') = G[V[S']]$. We also denote by $[v_i, v_j]$ the set $\{v_{i+1}, v_{i+2}, \ldots, v_{j-1}\}$ ($i < j$ in the ordering assumed for S [2]) such that: (a) for any $\ell = i, i+1, \ldots, j-1$, $(v_\ell, v_{\ell+1}) \in S$ (i.e., $[v_i, v_j]$ is the set of vertices in the path linking v_i to v_j in S, where v_i and v_j themselves are not encountered [3]). By symmetry, always for $i < j$, we denote by $[v_j, v_i]$ the set $\{v_{j+1}, v_{j+2}, \ldots, v_n, v_1, \ldots, v_{i-1}\}$. Obviously, $[v_i, v_j]$ and $[v_j, v_i]$ are both non-empty if S is a cordless cycle; $[v_j, v_i]$ is empty if S is a path or a tree.

Theorem 2 *Consider a problem Π verifying the following assumptions: (i) instances of Π are edge-valued complete graphs $(K_n, \vec{\ell}) = G(V, E, \vec{\ell})$; (ii) a solution of Π is a subset S of E inducing either a path, or a tree, or a cordless cycle; (iii) given an anticipatory solution S (the vertices of which are ordered in some appropriate order), algorithm A computes a feasible solution S', for any subgraph $G'(V', E', \vec{\ell}) = G[V']$ of G (obviously, G' is complete); (iv) $m(G, S) = \sum_{(v_i,v_j)\in S} \ell(v_i, v_j)$. Then, $E(G, S)$ is computable in polynomial time and is expressed by:*

$$E(G,S) = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)\, p_i p_j + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j)\, p_i p_j \prod_{v_l\in[v_i,v_j]} (1-p_l) + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j)\, p_i p_j \prod_{v_l\in[v_j,v_i]} (1-p_l)$$

Proof. Denote by $C[E']$, the set of edges added to S'' during the execution of step 2 of A. Obviously, $S' = S'' \cup C[E']$; also, if an edge belongs to $C[E']$, then it necessarily belongs to $E[V[S]]$, the set of edges of G induced by the endpoints of the edges in S. By assumptions (i) to (iii), S' is a feasible set of edges. Furthermore:

$$m(G',S') = \sum_{(v_i,v_j)\in E} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in S'\}} = \sum_{(v_i,v_j)\in E} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in S''\cup C[E']\}} \qquad (14)$$

By construction, any element of $C[E']$ is an edge (or arc) whose the initial endpoint corresponds to the terminal endpoint of a connected subset C_i of S, and the terminal endpoint corresponds to the initial endpoint of the "next" connected subset C_{i+1} of S. Then, for any subgraph G' of G, the following two assertions hold: (a) $S' \subseteq E''$, and (b) any edge that does not belong to E'', will never be part of any feasible solution (indeed, for such an edge, at least one of its endpoints does not belong to $V[S']$). So, $C[E'] \subseteq E''$. Then, from (14):

$$\begin{aligned} m(G',S') &= \sum_{(v_i,v_j)\in E} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in S''\cup C[E']\}} \\ &= \sum_{(v_i,v_j)\in E''} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in S''\cup C[E']\}} \\ &= \sum_{(v_i,v_j)\in E''} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in S''\}} + \sum_{(v_i,v_j)\in E''} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in C[E']\}} \\ &= \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in E'\}} + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in C[E']\}} \end{aligned} \qquad (15)$$

Using (15), we get from (1):

$$\begin{aligned} E(G,S) &= \sum_{V'\subseteq V} \left(\sum_{(v_i,v_j)\in S} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in E'\}} + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j)\, 1_{\{(v_i,v_j)\in C[E']\}} \right) \Pr[V'] \\ &= \sum_{(v_i,v_j)\in S} \ell(v_i,v_j) \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in E'\}} \Pr[V'] \\ &\quad + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j) \times \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in C[E']\}} \Pr[V'] \end{aligned} \qquad (16)$$

[2] Recall that S is either a path, or a tree, or a cordless cycle.

[3] It is assumed that if $[v_i, v_j] = \emptyset$, then $\prod_{v_l\in[v_i,v_j]}(1-p_l) = 0$.

As in the proof of Proposition 3, the first term of (16) can be simplified as follows:

$$\sum_{(v_i,v_j)\in S} \ell(v_i,v_j) \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in E'\}} \Pr[V'] = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)\, p_i p_j \tag{17}$$

Using (17) in (16), we get:

$$E(G,S) = \sum_{(v_i,v_j)\in S} \ell(v_i,v_j)\, p_i p_j + \sum_{(v_i,v_j)\in E''\setminus S} \ell(v_i,v_j) \sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in C[E']\}} \Pr[V'] \tag{18}$$

We now settle the second term of (18) that, in this form, seems to be exponential. Consider some edge (v_i, v_j) added during step 2 in order to "patch", say, connected components C_l and C_{l+1} of the anticipatory solution S. Since $(v_i, v_j) \notin S$, there exists in S a sequence $\mu = [v_i, v_j]$ of consecutive edges (or arcs) linking v_i to v_j. Assume that this sequence is listed by its vertices and that neither v_i, nor v_j belong to μ. Edge $(v_i, v_j) \in E'' \setminus S'$ is added to S' just because all the vertices in μ are absent. In other words, inclusion of (v_i, v_j) in $C[E']$ holds for any subgraph $G'(V', E')$, with $V' \in \mathcal{U}'_{ij} = \{V' \subseteq V : v_i \in V', v_j \in V'$ and any vertex of $\mu = [v_i, v_j]$ is absent$\}$. Consequently, the inner sum in the second term of (18) can be written as:

$$\sum_{V'\subseteq V} 1_{\{(v_i,v_j)\in C[E']\}} \Pr[V'] = \sum_{V'\in\mathcal{U}'_{ij}} \Pr[V'] = p_i p_j \prod_{v_l\in[v_i,v_j]} (1-p_l) + p_i p_j \prod_{v_l\in[v_j,v_i]} (1-p_l) \tag{19}$$

Combination of (16), (18) and (19) derives the expression claimed for the functional. It is easy to see that computation of a single term in the second sum of the functional requires $O(n)$ computations (at most $n+1$ multiplications). Since this is done at most $O(n^2)$ times (the edges in E), it follows that $E(G,S)$ is computable in $O(n^3)$, that concludes the proof of the theorem. ■

The fact that $E(G,S)$ is polynomial is partly due to the same reasons as in Propositions 1 and 3 and also to the way the "patching edges" are chosen at step 2 of A. Indeed, they are chosen in such a way that one can say a priori under which conditions an edge (or arc) (v_i, v_j) will be added in S'. These conditions carry over, the presence or the absence of the edges initially lying between v_i and v_j in S.

Unfortunately, in the opposite of Propositions 1 and 3, Theorem 2 does not derive a compact characterization for the optimal anticipatory solutions of the problems meeting the assumptions (i) to (iv). In particular, the form of the functional does not imply solution of some well-defined weighted version of Π (the deterministic support of PROBABILISTIC Π). This is due to the second term of the expression for $E(G,S)$ in Theorem 2. There, the "costs" assigned to the edges depend on the structure of the anticipatory solution chosen and of the present subgraph of G.

However, according to the functional in Theorem 2, we can easily conclude that when Π is **NP**-hard, so is PROBABILISTIC Π. In fact setting $p_i = 1$, for any $v_i \in V$, we recover the objective function of Π.

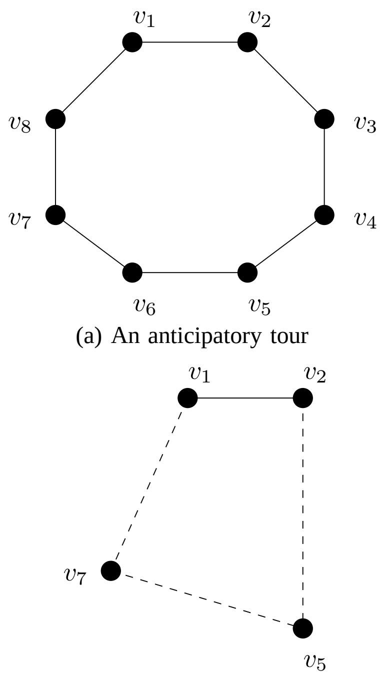
In what follows, we outline some problems fitting the conditions of Theorem 2. In particular, we study cases where feasible solutions are either cycles or trees.

6.1. *Application of Theorem 2 when the anticipatory solution is a cycle*

In this section, we consider MIN TSP and its probabilistic version. Given a complete graph on n vertices, denoted by K_n, with positive distances on its edges, MIN TSP consists of minimizing the cost of a Hamiltonian cycle (i.e., of an ordering $\langle v_1, v_2, \ldots, v_n\rangle$ of V such that $v_n v_1 \in E$ and, for $1 \leqslant i < n$, $v_i v_{i+1} \in E$), the cost of such a cycle being the sum of the distances of its edges. We shall represent any Hamiltonian cycle T (called also a tour in what follows) as the set of its edges; its value is $m(K_n, T) = \sum_{e\in T} \ell(v_i, v_j)$. Moreover, we arbitrarily number the vertices of K_n in the order that they are visited in T; so, we can set $T = \{(v_1, v_2), \ldots, (v_i, v_{i+1}), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}$.

Consider an anticipatory tour T in an edge-valued complete graph K_n and a set of absent vertices. Then, application of step 1 of A may result in a set $\{P_1, P_2, \ldots, P_k\}$ of paths [4], ordered in the order vertices have been visited in T, that is not feasible for MIN TSP in the surviving graph. In order to render this set feasible, one can link (modulo k) the last vertex of the path P_i to the first vertex of P_{i+1}; this is always possible since the initial graph is complete.

[4] These paths may be sets of edges, or simple edges, or even isolated vertices, any such vertex considered as a path.



(a) An anticipatory tour

(b) The tour T' computed by A

Fig. 2. An example of application of algorithm A for PROBABILISTIC MIN TSP.

For example, in Figure 2(a), an anticipatory cycle T, derived from a (symmetric) K_8 is shown. In Figure 2(b), we consider that vertices v_3, v_4, v_6 and v_8 are absent. In a first time, application of Step 1 of A results in a path-set $\{\{(v_1, v_2)\}, \{v_5\}, \{v_7\}\}$. In a second time, we will link vertex v_2 to v_5 (using the dotted edge (v_2, v_5)) and vertex v_5 to v_7 (by the dotted edge (v_5, v_7)). This creates a Hamiltonian path linking all the surviving vertices of the initial K_8. Finally, we link vertex v_7 to v_1 (by the dotted edge (v_7, v_1)). We so build a new tour feasibly visiting all the present vertices of the remaining graph.

It is easy to see that all the conditions of Theorem 2 are satisfied. Consequently, its application for the case of PROBABILISTIC MIN TSP gives for $E(K_n, T)$ the expression claimed in the theorem. We so recover the result of [9] about PROBABILISTIC MIN TSP. The anticipatory solution minimizing the functional cannot be characterized tightly by means of Theorem 2, since the expression for $E(K_n, T)$ depends on the particular anticipatory tour T considered and by the way this particular tour will be completed in the surviving instance.

6.2. *Application of Theorem 2 when the anticipatory solution is a tree*

Let us now consider MIN SPANNING TREE. Given an edge-valued graph $G(V, E, \vec{\ell})$, MIN SPANNING TREE consists of determining a tree T spanning V and minimizing quantity $m(G, T) = \sum_{e \in T} \ell(e)$. For the reasons discussed previously, we restrict ourselves to complete graphs.

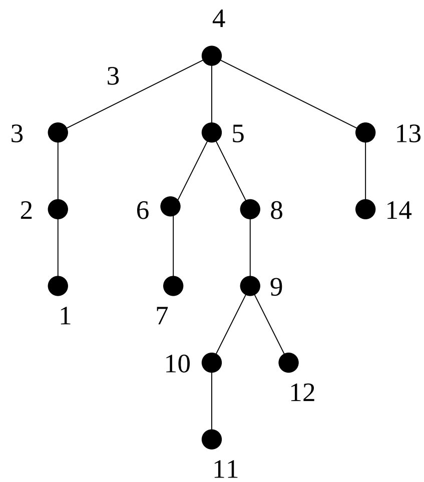
Note that in the case of PROBABILISTIC MIN TSP in Section 6.1., its solution induces an implicit and natural ordering of the edges. This is not the case here since various orderings can be considered. We consider a particular ordering of the vertices of T derived by a depth-first-search (dfs) starting from some leaf (numbered by 1). Obviously, this ordering is performed in $O(n)$ for a tree on n vertices (recall that such a tree has $n - 1$ edges).

For example, consider the tree of Figure 3(a) and assume that it is a minimum spanning tree of some graph on 14 vertices. In what follows vertices are named by their dfs number. This ordering partitions the edges of the tree into edge-disjoint paths $P_1, P_2, \ldots$ For instance, dealing with Figure 3(a), T is partitioned into 4 paths: $P_1 = \{1, 2, 3, 4, 5, 6, 7\}$, $P_2 = \{5, 8, 9, 10, 11\}$, $P_3 = \{9, 12\}$ and $P_4 = \{4, 13, 14\}$.

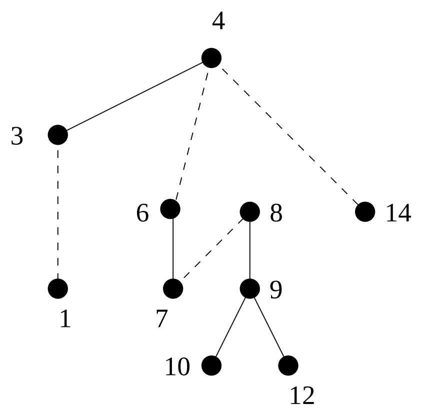
Suppose now that some vertices are absent from the initial graph G. Then, step 1 of A will produce a non connected set of edges (forming paths, any of them being a subset of some P_i); denote by $\{P'_1, P'_2, \ldots, P'_k\}$ the set of paths so-obtained. Order them according to the order of appearance of their edges appear in the dfs paths of T. For any $l = 1, \ldots, k$, we link the last vertex, say i of path P'_l to the first vertex, say j, of the path P'_{l+1}, if $i < j$. Since the initial graph is assumed complete, such an edge always exists.

With respect to the example of Figure 3(a), assume that vertices 2, 5, 11 and 13 disappear from the initial graph. Application of step 1 of A returns the following set of dfs paths: $P'_1 = \{1\}$, $P'_2 = \{3, 4\}$, $P'_3 = \{6, 7\}$, $P'_4 = \{8, 9, 10\}$, $P'_5 = \{9, 12\}$, $P'_6 = \{4\}$, $P'_7 = \{14\}$. The edges added in step 2 of A to reconnect the tree are $(1, 3)$, $(4, 6)$, $(7, 8)$, and $(4, 14)$ (Figure 3(b)). Note that the edges $(10, 9)$ connecting paths P'_4 and P'_5 and $(12, 4)$ connecting paths P'_5 and P'_6 have not been added since $10 > 9$ and $12 > 4$.

We now specify the path $[i, j]$ associated with the edge (i, j) connecting P'_l and P'_{l+1} and appearing in the expression for $E(G, S)$ in Theorem 2. Merging paths in the order they have been specified by the dfs numbering, T can be written as a sequence of vertices as



(a) The ordering of the nodes of an anticipatory solution T



(b) The solution T' derived from application of algorithm A on T

Fig. 3. When anticipatory solution is a tree.

they have been visited (of course, some of them appear more than once), i.e., $T = (1_1, 2_1, 3_1, \ldots, j_1, i_2, (j+1)_1, \ldots, (n-k)_q, n_1)$, where $i < j$ and i_c represents the c-th time the vertex i is encountered in T during the dfs. Based upon this representation, one can reconstruct T in the following way: for any pair $(i_c, j_{c'})$ of consecutive vertices, edge (i, j) belongs to T if and only if $i < j$. Note that a leaf appears only once in the list and that its absence does not disconnect the tree. Suppose now that some vertices are absent from the initial graph G. Drop them from the sequence representing T. This will produce a subsequence T' of T only including the copies of the present vertices. It is easy to see that the list T' is exactly the result of the concatenation of paths P'_i resulting from the removal of the absent vertices from the initial dfs paths. The reconnection of T performed by step 2 of A can be seen as insertion of an edge (i, j) linking two consecutive elements i_l and j_1 in T', where i_l is the last occurrence of i before the first occurrence j_1 of vertex j, verifying $i < j$. The corresponding path $[i, j]$ (i.e., the list of vertices that have to be absent in order that (i, j) is added), is the portion of the list between i_l and j_1.

Let us revisit the example of Figure 3(a). The sequence associated with the tree is $T = (1_1, \ldots, 7_1, 5_2, 8_1, \ldots, 11_1, 9_2, 12_1, 4_2, 13_1, 14_1)$ and, assuming that vertices 2, 5, 11 and 13 disappear, $T' = (1_1, 3_1, 4_1, 6_1, 7_1, 8_1, 9_1, 10_1, 9_2, 12_1, 4_2, 14_1)$. Then, $[1, 3] = \{2\}$, $[4, 6] = \{5\}$, $[7, 8] = \{5\}$ and $[4, 14] = \{13\}$.

By the discussion above, one can immediately conclude that $E(K_n, T)$ can be expressed as claimed by Theorem 2.

7. Final remarks

We have drawn a framework for the classification of probabilistic combinatorial optimization problems under the a priori optimization paradigm. What seems to be of interest in this classification is that when restriction of the initial solution to the "present" subgraph is feasible, then the complexity of determining the optimal anticipatory solution for the problems tackled, amounts to the complexity of solving some weighted version of the deterministic problem, where the weights depend on the vertex-probabilities. These weights do not depend on particular characteristics of the anticipatory solution considered, thing that allows a compact characterization of an optimal anticipatory solution. On the contrary, when more-than-one-stage algorithms are needed for building solutions, then the observation above is no more valid. In this case, one also recovers some weighted version of the original problem, but the weights on the data cannot be assigned independently of the structure of a particular anticipatory solution.

Acknowledgment. Many thanks to Orestis Telelis for very helpful discussions and pertinent comments on preliminary versions of the paper. The very useful comments of an anonymous referee are gratefully acknowledged.

References

[1] I. Averbakh, O. Berman, and D. Simchi-Levi. Probabilistic a priori routing-location problems. *Naval*

Res. Logistics, 41:973–989, 1994.

[2] M. Bellalouna, C. Murat, and V. Th. Paschos. Probabilistic combinatorial optimization problems: a new domain in operational research. *European J. Oper. Res.*, 87(3):693–706, 1995.

[3] D. J. Bertsimas. *Probabilistic combinatorial optimization problems*. Phd thesis, Operations Research Center, MIT, Cambridge Mass., USA, 1988.

[4] D. J. Bertsimas. On probabilistic traveling salesman facility location problems. *Transportation Sci.*, 3:184–191, 1989.

[5] D. J. Bertsimas. The probabilistic minimum spanning tree problem. *Networks*, 20:245–275, 1990.

[6] D. J. Bertsimas, P. Jaillet, and A. Odoni. A priori optimization. *Oper. Res.*, 38(6):1019–1033, 1990.

[7] L. Bianchi, J. Knowles, and N. Bowler. Local search for the probabilistic traveling salesman problem: correlation to the 2-p-opt and 1-shift algorithms. *European J. Oper. Res.*, 161(1):206–219, 2005.

[8] N. Bourgeois, F. Della Croce, B. Escoffier, C. Murat, and V. Th. Paschos. Probabilistic coloring of bipartite and split graphs. *J. Comb. Optimization*, 17(3):274–311, 2009.

[9] P. Jaillet. Probabilistic traveling salesman problem. Technical Report 185, Operations Research Center, MIT, Cambridge Mass., USA, 1985.

[10] P. Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Oper. Res.*, 36(6):929–936, 1988.

[11] P. Jaillet. Shortest path problems with node failures. *Networks*, 22:589–605, 1992.

Received 3-4-2009; revised 21-10-2009; accepted 25-11-2009

[12] P. Jaillet and A. Odoni. The probabilistic vehicle routing problem. In B. L. Golden and A. A. Assad, editors, *Vehicle routing: methods and studies*. North Holland, Amsterdam, 1988.

[13] C. Murat and V. Th. Paschos. The probabilistic longest path problem. *Networks*, 33:207–219, 1999.

[14] C. Murat and V. Th. Paschos. A priori optimization for the probabilistic maximum independent set problem. *Theoret. Comput. Sci.*, 270:561–590, 2002.

[15] C. Murat and V. Th. Paschos. The probabilistic minimum vertex-covering problem. *Int. Trans. Opl Res.*, 9(1):19–32, 2002.

[16] C. Murat and V. Th. Paschos. L'optimisation combinatoire probabiliste. In V. Th. Paschos, editor, *Optimisation combinatoire : concepts avancés*, chapter 6, pages 221–247. Hermès Science, Paris, 2005.

[17] C. Murat and V. Th. Paschos. On the probabilistic minimum coloring and minimum k-coloring. *Discrete Appl. Math.*, 154:564–586, 2006.

[18] C. Murat and V. Th. Paschos. *Probabilistic combinatorial optimization on graphs*. ISTE and Hermès Science Publishing, London, 2006.

[19] V. Th. Paschos, O. A. Telelis, and V. Zissimopoulos. Probabilistic models for the STEINER TREE problem. *Networks*. To appear.

[20] V. Th. Paschos, O. A. Telelis, and V. Zissimopoulos. Steiner forests on stochastic metric graphs. In A. Dress, Y. Xu, and B. Zhu, editors, *Proc. Conference on Combinatorial Optimization and Applications, COCOA'07*, volume 4616 of *Lecture Notes in Computer Science*, pages 112–123. Springer-Verlag, 2007.