Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Shortest Path Based Decision Making Using Probabilistic Inference

Akshat Kumar
School of Information Systems
Singapore Management University
akshatkumar@smu.edu.sg

Abstract

We present a new perspective on the classical shortest path routing (SPR) problem in graphs. We show that the SPR problem can be recast to that of probabilistic inference in a mixture of simple Bayesian networks. Maximizing the likelihood in this mixture becomes equivalent to solving the SPR problem. We develop the well known Expectation-Maximization (EM) algorithm for the SPR problem that maximizes the likelihood, and show that it *does not* get stuck in a locally optimal solution. Using the same probabilistic framework, we then address an NP-Hard *network design* problem where the goal is to repair a network of roads post some disaster within a fixed budget such that the connectivity between a set of nodes is optimized. We show that our likelihood maximization approach using the EM algorithm scales well for this problem taking the form of message-passing among nodes of the graph, and provides significantly better quality solutions than a standard mixed-integer programming solver.

1 Introduction

The shortest path routing (SPR) problem entails finding the shortest path between two given vertices of the graph that minimizes the total sum of *weights* of the involved edges in the path (Dijkstra 1959; Cormen et al. 2001). Shortest path based approaches have found several applications in diverse fields such as transportation models (Pallottino and Scutella 1998), telecommunication network design (Pioro et al. 2002) and in ecology for analyzing landscape connectivity for conservation planning (Bunn, Urban, and Keitt 2012; Minor and Urban 2008). The classical shortest path problem, which is tractable, is typically solved using dynamic programming based approaches such as Dijkstra's or Floyd–Warshall's algorithm (Cormen et al. 2001) and their variants.

In our work, we take a different approach by developing a novel graphical models based probabilistic perspective on the SPR problem. Recently, there has been tremendous progress in variational approaches for inference in graphical models (Yanover et al. 2006; Sontag et al. 2008; Sontag and Jaakkola 2007; Wainwright and Jordan 2008). Our goal is to show how the SPR problem (more importantly, its NP-Hard variants) can be reformulated as a likelihood maximization inference problem in an appropriately constructed mixture of simple Bayesian networks. We can then use several existing approaches for likelihood maximization (LM) such as the well known expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) and other variational inference approaches (Liu and Ihler 2013) for solving shortest path based decision making (SPDM) problems. We also address the crucial issue of extracting the integral solution for SPDM problems from their LM-based continuous solution. We propose an entropy-based penalty term that encourages deterministic solutions within the LM framework and eliminates the need for ad-hoc approaches for rounding the continuous solution. We show that the resulting entropy-augmented LM can be solved using the difference-of-convex functions (DC) programming approach (Yuille and Rangarajan 2001). To summarize, our work introduces a promising new framework which combines SPDM with probabilistic inference, and opens the door to the application of rich inference and optimization-based techniques to solve SPDM problems.

Recently, there is an increasing interest in solving sequential decision making problems under uncertainty, such as Markov decision processes (MDPs), partially observable MDPs (POMDPs) and its multiagent variants using the probabilistic inference based viewpoint (Toussaint and Storkey 2006; Toussaint, Charlin, and Poupart 2008; Kumar and Zilberstein 2010; Ghosh, Kumar, and Varakantham 2015; Kumar, Zilberstein, and Toussaint 2015). Our approach for SPDM is different from such previous applications of the LM framework for planning. Most of the previous planning-as-inference approaches work for infinite horizon problems with reward discounting, whereas in the SPDM problem, there is no future cost discounting. Previous inference-based approaches do computation (in the form of message-passing) on a time-indexed representation of the problem to take into account the sequential nature of (PO)MDP models (Toussaint, Charlin, and Poupart 2008). However, explicitly taking into account the number of planning steps in SPDM problems would make the underlying graph extremely large leading to computational intractability as SPDM problems have indefinite-horizon (i.e., the length of the shortest path is not known a priori). The *network flow* constraints that arise in SPDM problems also require different set of techniques than previous approaches.

As a concrete instance of an NP-Hard SPDM problem, we address a road network design problem (RNDP) where the goal is to repair a network of roads post some disaster within a fixed budget such that the connectivity between a given set of nodes is optimized. Several different variants of this problem have received attention recently (Aksu and Ozdamar 2014; Ozdamar, Aksu, and Ergunes 2014; Liberatore et al. 2014; Duque and Sorensen 2011). Aksu and Ozdamar [2014] address the road restoration problem by identifying a set of critical edges from the *predefined* set of paths that need to be restored with limited resources. Similarly, Liberatore et al. address the road repair problem with the aim of optimizing humanitarian relief distribution. Duque and Sorensen address a similar problem of repairing a rural road network post some disaster. Our work differs from previous approaches in several ways. We do not predefine the set of paths that need to cleared (as in (Aksu and Ozdamar 2014)), instead we simultaneously optimize the road repair decisions and shortest paths that depend on road repair decisions to optimize connectivity. Most previous approaches for RNDP are based on mixed-integer programming (MIP) (Aksu and Ozdamar 2014; Liberatore et al. 2014) or local search based heuristics (Duque and Sorensen 2011; Ozdamar, Aksu, and Ergunes 2014). In contrast, we directly solve a nonlinear formulation of the problem based on our LM framework that is highly competitive with MIP solvers for small/moderate sized problems, and significantly outperforms them w.r.t. solution quality for larger instances.

2 Shortest Path As Probabilistic Inference

Consider a directed graph $G=(V,E)$. Nodes in this graph are denoted using $i \in V$, and directed edges $(i,j) \in E$. Associated with each edge is a weight $w_{ij} \in \Re^+$. We are interested in finding the shortest path (as per the weights w_{ij}) from source node s to destination node t. We write below the standard LP formulation for the shortest path:

$$\min_{\boldsymbol{x}} \sum_{(i,j)\in E} w_{ij} x_{ij} \tag{1}$$

$$\sum_{j\in \text{Nb}_i} x_{ij} - \sum_{j\in \text{Nb}_i} x_{ji} = \begin{cases} 1 \text{ if } i=s; \\ -1 \text{ if } i=t; \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in V \tag{2}$$

$$x_{ij} \in [0,1] \ \forall (i,j) \in V \tag{3}$$

where Nb_i denotes the neighbors of a node i. Constraints (2) are referred to as *flow constraints*.

We now reformulate the problem of finding the shortest path between source s and destination t as LM in a mixture of simple Bayesian networks (BN). We create one BN for each directed edge (i,j). It has two binary random variables $l_{ij} \in \{0,1\}$ and $r \in \{0,1\}$. The variable l_{ij} is the parent of the variable r. Figure 1(a) shows a graph with 5 nodes and 10 directed edges, figure 1(b) shows Bayes nets for different edges. The mixture variable is L, whose domain is the set of all edges E, and has a *fixed* uniform distribution $\Pr(L=(i,j)) = \frac{1}{|E|}$. For brevity, the assignment $L=(i,j)$ is denoted as L_{ij}. Let $w^\star = w_{\max}+1$ where $w_{\max}$ denotes the

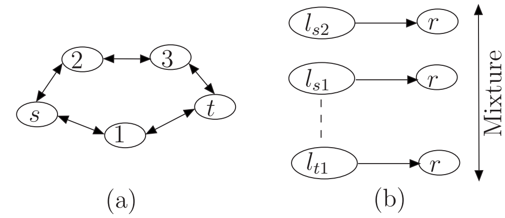


Figure 1: Mixture model corresponding to a graph

maximum weight of any edge in the graph. The CPT of the variable r for the Bayes net corresponding to L_{ij} is set as:

$$\Pr(r=1 \mid l_{ij}=1) = \frac{w^\star - w_{ij}}{w^\star} \tag{4}$$

$$\Pr(r=1 \mid l_{ij}=0) = 1 \tag{5}$$

Let the prior for variables l_{ij} be denoted by $\tilde{x}_{ij} = \Pr(l_{ij}=0)$ and $x_{ij} = \Pr(l_{ij}=1)$. We therefore also have a relation that $x_{ij} + \tilde{x}_{ij} = 1$. Intuitively, parameters $x_{ij} = \Pr(l_{ij}=1)$ have the same interpretation as the x variables in the constraint (2) of the shortest path LP.

Theorem 1. *Let the CPT of binary variable r in the mixture model be set as per* (4) *and* (5)*, and parameters $\boldsymbol{x} = \{x_{ij}, \tilde{x}_{ij} \forall (i,j)\}$ satisfy the flow constraints* (2)*, then maximizing the likelihood $P(r=1;\boldsymbol{x})$ of observing $r=1$ in the mixture model is equivalent to solving the SPR problem.*

Proof. The full joint for the BN mixture is given as:

$$P(r=1;\boldsymbol{x}) = \sum_{(i,j)} P(r=1, L=(i,j)) \tag{6}$$

$$= \sum_{(i,j)} \big[P(r=1, l_{ij}=0, L_{ij}) + P(r=1, l_{ij}=1, L_{ij})\big] \tag{7}$$

$$\propto \sum_{(i,j)} \big[1\cdot \tilde{x}_{ij} + \frac{w^\star - w_{ij}}{w^\star} x_{ij}\big] \tag{8}$$

$$= \sum_{(i,j)} \frac{w^\star \tilde{x}_{ij} + w^\star x_{ij} - w_{ij} x_{ij}}{w^\star} \tag{9}$$

$$= \sum_{(i,j)} \frac{w^\star - w_{ij} x_{ij}}{w^\star} = |E| - \frac{1}{w^\star} \sum_{(i,j)} w_{ij} x_{ij} \tag{10}$$

Therefore, we have the following relation:

$$\max_{\boldsymbol{x}} P(r=1;\boldsymbol{x}) \propto |E| - \min_{\boldsymbol{x}} \frac{1}{w^\star} \sum_{(i,j)} w_{ij} x_{ij} \tag{11}$$

Thus maximizing the likelihood is equivalent to minimizing the objective of the shortest path LP in (1). □

3 Finding Shortest Path Using EM

The EM algorithm is a general approach to the problem of maximum likelihood parameter estimation in models with latent variables (Dempster, Laird, and Rubin 1977). Thm. 1 provides a clear connection for applying the EM algorithm for SPR. In our mixture model, all the variables ($\boldsymbol{l}$, L) are hidden. Only the variable $r = 1$ is observed. Our goal is to find the best parameters $\boldsymbol{x}^{\text{opt}}$ that maximize the log-likelihood below:

$$\boldsymbol{x}^{\text{opt}} = \arg\max_{\boldsymbol{x}} \log\left(|E| - \frac{1}{w^\star}\sum_{(i,j)} w_{ij}x_{ij}\right) \quad (12)$$

subject to the *flow constraints* (2) on parameters x_{ij} and the *normalization constraints* $x_{ij} + \tilde{x}_{ij} = 1$. The EM algorithm is an iterative approach that performs coordinate ascent in the parameter space. In each iteration, EM maximizes the following function, also called *expected complete log-likelihood* $Q(\boldsymbol{x}, \boldsymbol{x}^\star)$:

$$\propto \sum_{(i,j), l_{ij}\in\{0,1\}} P(r=1, l_{ij}, L_{ij}; \boldsymbol{x}) \log P(r=1, l_{ij}, L_{ij}; \boldsymbol{x}^\star) \quad (13)$$

where $\boldsymbol{x}$ denote last iteration's parameters and $\boldsymbol{x}^\star$ denote the parameters to be optimized. We can further simplify the function Q as below:

$$\sum_{(i,j)\in E} \Big[P(r=1, l_{ij}=0, L_{ij}; \boldsymbol{x}) \log P(r=1, l_{ij}=0, L_{ij}; \boldsymbol{x}^\star)$$

$$+ P(r=1, l_{ij}=1, L_{ij}; \boldsymbol{x}) \log P(r=1, l_{ij}=1, L_{ij}; \boldsymbol{x}^\star) \Big] \quad (14)$$

$$\propto \sum_{(i,j)\in E} \left[\tilde{x}_{ij} \log \tilde{x}^\star_{ij} + \frac{w^\star - w_{ij}}{w^\star} x_{ij} \log x^\star_{ij} \right] \quad (15)$$

$$\propto \sum_{(i,j)\in E} \left[\tilde{x}_{ij} \log \tilde{x}^\star_{ij} + \hat{w}_{ij} x_{ij} \log x^\star_{ij} \right] \quad (16)$$

where we used $\hat{w}_{ij} = \frac{w^\star - w_{ij}}{w^\star}$ to denote normalized edge weights. In the above expressions, we have also ignored terms that are independent of parameters $\boldsymbol{x}^\star$ as they do not affect the optimization w.r.t. $\boldsymbol{x}^\star$. We next show that EM would converge to the global optimum of the SPR problem and will not get stuck in a local optima.

Theorem 2. *The EM algorithm for maximizing the likelihood of $r=1$ in the SPR mixture model would converge to a global optimum of the log-likelihood.*

Proof. EM algorithm converges to a *stationary point* of the log-likelihood function if the expected log-likelihood Q is continuous in both the parameters $\boldsymbol{x}$ and $\boldsymbol{x}^\star$ (Wu 1983). The Q function for our case in (16) satisfies this condition. Therefore, EM algorithm would converge to the stationary point of log-likelihood. As our log-likelihood function (12) is concave (and flow constraints (2) linear), the stationary point is also a global optima (Bertsekas 1999). □

3.1 Maximizing the Expected Log-Likelihood Q

We now detail the procedure to maximize the expected log-likelihood function Q in (16) (note the sign change below).

$$\min_{x^\star_{ij}, \tilde{x}^\star_{ij}} -\sum_{(i,j)} \tilde{x}_{ij} \log \tilde{x}^\star_{ij} - \sum_{(i,j)} \hat{w}_{ij} x_{ij} \log x^\star_{ij} \quad (17)$$

$$\text{s.t.} \sum_{j\in \text{Nb}_i} x^\star_{ji} - \sum_{j\in \text{Nb}_i} x^\star_{ij} + k_i = 0 \; \forall i \in V \quad (18)$$

$$\tilde{x}^\star_{ij} + x^\star_{ij} = 1 \; \forall (i,j) \in E;\; \tilde{x}^\star_{ij}, x^\star_{ij} \in [0,1] \quad (19)$$

where the value of the constant $k_i \in \{-1, 0, 1\}$ depends upon whether the node is source s, destination t or any other node. The above problem does not admit a closed form solution. Therefore, we use several tools from convex optimization and algebra to develop a graph-based scalable message-passing algorithm. Our high level approach is as follows:

- We write the Lagrangian dual of problem (17). The dual has simpler structure making optimization easier. Furthermore, as (17) is a convex optimization problem, there is no duality gap implying optimal dual quality equals optimal of (17) (Bertsekas 1999).
- To optimize the dual, we use results from convex optimization (Bertsekas 1999) that guarantee that a block coordinate ascent (BCA) strategy wherein we fix all the dual variables except one, and then optimize the dual over the one variable is guaranteed to converge to the optimal dual solution.

Dual of (17) We first define the Lagrangian function as:

$$L(\boldsymbol{x}^\star, \boldsymbol{\lambda}) = -\sum_{(i,j)} \tilde{x}_{ij} \log \tilde{x}^\star_{ij} - \sum_{(i,j)} \hat{w}_{ij} x_{ij} \log x^\star_{ij} +$$

$$\sum_i \lambda_i \Big[\sum_{j\in \text{Nb}_i} x^\star_{ji} - \sum_{j\in \text{Nb}_i} x^\star_{ij} + k_i \Big] + \sum_{ij} \lambda_{ij} \left[\tilde{x}^\star_{ij} + x^\star_{ij} - 1 \right]$$

where $\boldsymbol{\lambda} = \{\lambda_i \forall i, \lambda_{ij} \forall (i,j)\}$ include dual variables λ_i for the flow conservation constraint for a node i, and λ_{ij} for the normalization constraint. The dual function is given as $q(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}^\star} L(\boldsymbol{x}^\star, \boldsymbol{\lambda})$. This function is found by setting the partial derivative of $L(\boldsymbol{x}^\star, \boldsymbol{\lambda})$ w.r.t. $x^\star_{ij}$ and $\tilde{x}^\star_{ij}$ to zero. Upon simplifying, we get the dual function as:

$$q(\boldsymbol{\lambda}) = \sum_{(i,j)} \tilde{x}_{ij} \log \lambda_{ij} + \sum_{(i,j)} \hat{w}_{ij} x_{ij} \log\left(\lambda_j + \lambda_{ij} - \lambda_i\right) +$$

$$\sum_i k_i \lambda_i + \sum_{ij} \hat{w}_{ij} x_{ij} + \sum_{ij} \tilde{x}_{ij} - \sum_{ij} \lambda_{ij} + \text{const. terms} \quad (20)$$

Maximizing the Dual (20) It is a standard result in convex optimization that for any value of dual variables $\boldsymbol{\lambda}$, $q(\boldsymbol{\lambda}) \le Q^{\text{opt}}$, where Q^{opt} denotes the optimal value of (17). Therefore, we now detail how to maximize the dual $q(\boldsymbol{\lambda})$. We use the block-coordinate ascent (BCA) strategy to optimize the dual. We choose an arbitrary dual variable, say λ_i, fix all the other dual variables and optimize the function $q(\cdot)$ w.r.t. the chosen variable λ_i. In general, this strategy is not guaranteed to converge to the optimal solution. However, the function $q(\boldsymbol{\lambda})$ satisfies additional properties which guarantee that the BCA approach will converge to the optimal dual solution. These conditions are 1) $q(\cdot)$ is continuously differentiable over its domain; 2) $q(\cdot)$ is strictly concave w.r.t. each dual variable λ_i and λ_{ij} due to the presence of log terms in (20), resulting in a unique solution for each BCA iteration (Bertsekas 1999).

Maximizing the Dual (20) **w.r.t.** λ_i The optimization problem to solve is $\max_{\lambda_i} q(\boldsymbol{\lambda})$. We set the partial derivative $\frac{\partial q}{\partial \lambda_i}$ to zero and get the condition:

$$f(\lambda_i) = \sum_{j\in \text{Nb}_i} \frac{\hat{w}_{ji} x_{ji}}{\lambda_i + (\lambda_{ji} - \lambda_j)} - \sum_{j\in \text{Nb}_i} \frac{\hat{w}_{ij} x_{ij}}{(\lambda_j + \lambda_{ij}) - \lambda_i} + k_i = 0 \quad (21)$$

The above equation has many solutions as it is a linear combination of rational functions in λ_i. This appears problematic as we need a unique solution for λ_i as required by the BCA approach. Fortunately, we present the analysis below that shows us that there is precisely one root of the above equation that will satisfy our requirements. First notice the $\log$ terms in the dual (20). As $\log$ argument must always be positive (so that dual is not $-\infty$), we require the root of (21) that satisfies the following conditions simultaneously:

$$\lambda_i > \lambda_j - \lambda_{ji} \; \forall j \in \text{Nb}_i \quad \Rightarrow \; \lambda_i > \max_{j \in \text{Nb}_i} \left(\lambda_j - \lambda_{ji}\right) \tag{22}$$

$$\lambda_j + \lambda_{ij} - \lambda_i > 0 \; \forall j \in \text{Nb}_i \; \Rightarrow \; \lambda_i < \min_{j \in \text{Nb}_i} \left(\lambda_j + \lambda_{ij}\right) \tag{23}$$

Conditions (22) and (23) are the required conditions. Essentially, the value $\max_{j \in \text{Nb}_i} \left(\lambda_j - \lambda_{ji}\right)$ denotes the lower bound for λ_i and $\min_{j \in \text{Nb}_i} \left(\lambda_j + \lambda_{ij}\right)$ denotes the upper bound. We denote these values as:

$$\lambda_i^{\min} = \max_{j \in \text{Nb}_i} \left(\lambda_j - \lambda_{ji}\right) \text{ and } \lambda_i^{\max} = \min_{j \in \text{Nb}_i} \left(\lambda_j + \lambda_{ij}\right) \tag{24}$$

We also observe that discontinuities in the function $f(\lambda_i)$ (21) occur at $\lambda_i = (\lambda_j - \lambda_{ji}) \forall j \in \text{Nb}_i$ and $\lambda_i = \lambda_j + \lambda_{ij} \forall j \in \text{Nb}_i$. We now state the following proposition.

Proposition 1. *Assuming that $\lambda_i^{\min} < \lambda_i^{\max}$, the function $f(\lambda_i)$ in* (21) *has exactly one root in the open interval $\left(\lambda_i^{\min}, \lambda_i^{\max}\right)$.*

Proof. Notice that the first summation in (21), $\sum_{j \in \text{Nb}_i} \frac{\hat{w}_{ji} x_{ji}}{\lambda_i + (\lambda_{ji} - \lambda_j)}$, does not cause any discontinuity for any $\lambda_i > \lambda_i^{\min}$ as denominators would never be zero for any such λ_i. Similarly, the second summation term, $\sum_{j \in \text{Nb}_i} \frac{\hat{w}_{ij} x_{ij}}{(\lambda_j + \lambda_{ij}) - \lambda_i}$, does not cause any discontinuity for any $\lambda_i < \lambda_i^{\max}$. Therefore, given that $\lambda_i^{\min} < \lambda_i^{\max}$, we can deduce that $f(\lambda_i)$ is continuous in the interval $\left(\lambda_i^{\min}, \lambda_i^{\max}\right)$. We also observe that the function $f(\lambda_i)$ is monotonically decreasing in the interval $\left(\lambda_i^{\min}, \lambda_i^{\max}\right)$. This can be verified by checking the first derivative of $f(\lambda_i)$, which is always negative in this interval (all numerators in $f(\lambda_i)$ are positive).

Consider interval $\left[\lambda_i^{\min} + \epsilon, \lambda_i^{\max} - \epsilon\right]$ for any $\epsilon > 0$. We have

$$f(\lambda_i^{\min} + \epsilon) = \frac{k_a}{\epsilon} + \ldots \text{ and } f(\lambda_i^{\max} - \epsilon) = \frac{-k_b}{\epsilon} + \ldots$$

where k_a and k_b are positive numbers. Therefore, we have the condition that as $\epsilon \to 0$, $f(\lambda_i^{\min} + \epsilon) \to \infty$ and $f(\lambda_i^{\max} - \epsilon) \to -\infty$. Since, we know that $f(\lambda_i)$ is also continuous and monotonically decreasing in the interval $\left(\lambda_i^{\min}, \lambda_i^{\max}\right)$, it must cross the horizontal axis $y = 0$ exactly once. This completes our proof. □

The above proposition provides the solution to our problem. We know from conditions (22) and (23) that λ_i must lie in the interval $\left(\lambda_i^{\min}, \lambda_i^{\max}\right)$, and prop. 1 shows that there is exactly one root in this interval. Therefore, this root must be the solution to be used for the BCA iteration. We can find this root by using one of the many root finding techniques, such as the Brent's method (Brent 1971).

Algorithm 1: pathEM$(G = (V, E), s, t)$

Initialize: $x_{ij} \leftarrow 0.5$; $\tilde{x}_{ij} \leftarrow 0.5 \; \forall (i,j) \in E$
repeat
  Set $\lambda_i \leftarrow 0 \; \forall i \in V$; $\lambda_{ij} \leftarrow 1 \; \forall (i,j) \in E$
  repeat
    for *each edge* $(i,j) \in E$ **do**
      Find *unique largest* root λ'_{ij} for $g(\lambda'_{ij}) = 0$:
      $g(\lambda'_{ij}) = \frac{\tilde{x}_{ij}}{\lambda'_{ij}} + \frac{\hat{w}_{ij} x_{ij}}{\lambda_j + \lambda'_{ij} - \lambda_i} - 1$
      $\lambda_{ij} \leftarrow \lambda'_{ij}$
    for *each node* $i \in V$ **do**
      Find *unique* root $\lambda'_i \in (\lambda_i^{\min}, \lambda_i^{\max})$ for $f(\lambda'_i) = 0$:
      $f(\lambda'_i) = \sum_{j \in \text{Nb}_i} \frac{\hat{w}_{ji} x_{ji}}{\lambda'_i + (\lambda_{ji} - \lambda_j)} - \sum_{j \in \text{Nb}_i} \frac{\hat{w}_{ij} x_{ij}}{(\lambda_j + \lambda_{ij}) - \lambda'_i} + k_i$
      $\lambda_i \leftarrow \lambda'_i$
  until ***convergence***
  $x^\star_{ij} \leftarrow \frac{\hat{w}_{ij} x_{ij}}{\lambda_j + \lambda_{ij} - \lambda_i}$, $\tilde{x}^\star_{ij} \leftarrow \frac{\tilde{x}_{ij}}{\lambda_{ij}} \; \forall (i,j) \in E$
  $x_{ij} \leftarrow x^\star_{ij}$, $\tilde{x}_{ij} \leftarrow \tilde{x}^\star_{ij} \; \forall (i,j) \in E$
until ***convergence***
return Extracted path (from s to t) from $\boldsymbol{x}$ variables

The only remaining thing to show is that after every update of λ_i, our invariant condition $\lambda_j^{\min} < \lambda_j^{\max}$ is maintained for each $\lambda_j \; \forall j \in \text{Nb}_i$ as the update of λ_i only affects the invariant conditions of its immediate neighbors. The proposition next shows this result.

Proposition 2. *Let the current estimate of the dual variables be denoted as λ_i, λ_j and λ_{ij}. Once λ_i gets updated to λ'_i using equation* (21)*, we have for every $j \in Nb_i$:*

$$\max\left(\lambda'_i - \lambda_{ij}, \max_{k \in Nb_j \setminus i} \left(\lambda_k - \lambda_{kj}\right)\right) < \min\left(\lambda'_i + \lambda_{ji}, \min_{k \in Nb_j \setminus i} \left(\lambda_k + \lambda_{jk}\right)\right)$$

Proof is provided in the supplement.

Maximizing the Dual (20) **w.r.t.** λ_{ij} The optimization problem to solve is $\max_{\lambda_{ij}} q(\boldsymbol{\lambda})$. Its analysis is similar to the one presented for variable λ_i. We can also prove an analogue of the proposition 2 by showing that variables λ_i and λ_j (which are the only ones affected by the update of λ_{ij}) satisfy their respective invariant conditions after the update of λ_{ij} variable.

We summarize all the steps in the algorithm 1. The EM algorithm takes the form of a double-loop algorithm. The outer loop corresponds to EM's iterations, and inner loop corresponds to BCA approach's iterations to maximize the dual. Upon convergence, the variables $\boldsymbol{x}$ are close to integral, and a path from source to destination can be extracted from it. The convergence of the inner loop is detected via measuring the violations of the flow constraints and the probability normalization constraints. The convergence of the outer loop is detected if the increase in quality is less

$$\min_{\boldsymbol{y},\{\boldsymbol{x}^m\forall m\}} \sum_{m=1}^{M}\sum_{(i,j)\in E}\sum_{a_{ij}} y_{a_{ij}} w_{a_{ij}} x_{ij}^m \tag{25}$$

$$\sum_{a_{ij}} y_{a_{ij}} = 1 \;\forall (i,j) \tag{26}$$

$$\sum_{(i,j)}\sum_{a_{ij}} c_{a_{ij}} y_{a_{ij}} \leq B \tag{27}$$

$$\sum_{j\in \text{Nb}_i} x_{ij}^m - \sum_{j\in \text{Nb}_i} x_{ji}^m = \begin{cases} 1 \text{ if } i = o_m; \\ -1 \text{ if } i = d_m; \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in V\ , \forall m \tag{28}$$

$$x_{ij}^m \in [0,1] \;\forall (i,j) \in V,\; \forall m \tag{29}$$

$$y_{a_{ij}} \in \{0,1\} \;\forall (i,j) \tag{30}$$

Table 1: Mixed-integer quadratic program for RNDP

than a particular threshold. Notice that all the updates EM requires can be implemented using message-passing on the graph G in a distributed manner, which is useful for applications such as multiagent path finding. The complexity of each inner loop iteration is linear in the number of edges of the graph, thus, EM can be scaled to large graph sizes.

4 Network Design Using LM

One main benefit of developing a graphical models and likelihood maximization based perspective on the SPR problem is that this reasoning easily extends to cases when the objective function to optimize is nonlinear and nonconvex, and the problem NP-Hard. Such problems may be hard to handle using classical algorithms such as Dijkstra's, but as we show later, the EM algorithm still applies with relatively little modifications. We next discuss a road network design problem (RNDP) where the goal is to repair a network of roads post some disaster within a fixed budget such that the connectivity between a given set of nodes is optimized.

The RNDP problem is specified by the tuple $\langle G, \text{odList}, \boldsymbol{A}, \boldsymbol{W}, \boldsymbol{C}, B\rangle$. We have a road network as a directed graph $G = (V, E)$. The set $\text{odList} = \{(o_m, d_m)\}$ consists of M different origin-destination pairs (o_m, d_m). We assume that different roads are damaged to different extents. To repair a segment (i,j), possible actions are denoted using the set A_{ij}. The joint-set of all possible actions is $\boldsymbol{A}$. If a repair action $a_{ij} \in A_{ij}$ is performed, then the *edge weight* of the link (i,j) is given using $W_{ij}(a_{ij})$. The *cost* of a repair action a_{ij} is given using $C_{ij}(a_{ij})$. Intuitively, a higher cost action would lead to lower edge weight. We assume that a default action *noop* is included for each edge which has a zero cost and some default edge weight. We are also given a budget B that limits the quality of repair actions for different edges. Let $\boldsymbol{y} = \{y_{a_{ij}} \forall a_{ij} \in A_{ij} \forall (i,j)\}$ denote a binary decision vector; $y_{a_{ij}} = 1$ implies action a_{ij} is taken for edge (i,j). Let $\text{SP}(o_m, d_m; \boldsymbol{y})$ denote the total

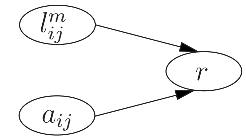


Figure 2: Single mixture component Bayes net corresponding to edge (i,j) and od pair m

edge weight of the shortest directed path from o_m to d_m as per the decision $\boldsymbol{y}$, then our goal is to find the best decision $\boldsymbol{y}^\star$ to minimize $\sum_{(o_m,d_m)\in\text{odList}} \text{SP}(o_m, d_m; \boldsymbol{y})$ such that the total cost of decision $\boldsymbol{y}^\star$ is less than the budget B. This problem is NP-Hard, which can be shown by reducing the Knapsack problem to it (proof omitted).

Table 1 shows a mixed-integer quadratic program (MIQP) for this problem. The main difference of this program from the shortest path LP in (1) is that the edge weight is variable (depends on the decision $y_{a_{ij}}$) and denoted using $\sum_{a_{ij}} y_{a_{ij}} w_{a_{ij}}$. Therefore, the objective becomes quadratic. Constraints of this MIQP include flow constraints (28) for each o-d pair m and the budget constraint (27). We used a shorthand of $c_{a_{ij}}$ to denote the cost of the decision a_{ij}, and $w_{a_{ij}}$ denotes the corresponding edge weight. To solve this MIQP, we first reformulate it as an MIP. We use the standard technique to linearize each term $y_{a_{ij}} \cdot x_{ij}^m$ by replacing it with $z_{a_{ij}}^m$, and adding three linear constraints as $z_{a_{ij}}^m \leq x_{ij}^m$, $z_{a_{ij}}^m \leq y_{a_{ij}}$ and $z_{a_{ij}}^m \geq x_{ij}^m + y_{a_{ij}} - 1$. Only the variables $\{y_{a_{ij}}\}$ are binary, the rest are continuous ($\in [0,1]$).

In our work, we propose a scalable alternative to MIP by solving a relaxed version of this MIQP using the LM framework and the EM algorithm. Our strategy is following: 1) We construct a mixture graphical model such that maximizing the likelihood of observing the variable $r = 1$ is equivalent to solving the relaxed QP in table 1 with decision variable $\boldsymbol{y}$ becoming *continuous*, 2) We develop the EM algorithm to maximize the likelihood in this mixture, 3) We incorporate an entropy-based term in the EM algorithm that encourages deterministic solutions resulting in integral final decision $\boldsymbol{y}$.

We start by creating a number of simple Bayes nets (as in figure 1) for each edge (i,j) and od pair m. Figure 2 shows the structure of a single Bayes net. Binary variable l_{ij}^m has same interpretation as in figure 1, its prior distribution is denoted using x_{ij}^m. In addition, we have another variable a_{ij} whose domain is the set of all available repair actions A_{ij} for the edge (i,j). For any $a_{ij} \in A_{ij}$, we have $\Pr(a_{ij}) = y_{a_{ij}}$. Defining priors in this way establishes the connection of LM to the QP variables $\boldsymbol{x}$ and $\boldsymbol{y}$ in table 1. We define the CPT of different variables in the BN as:

$$P(r=1|l_{ij}^m = 1, a_{ij}) = \hat{w}_{a_{ij}} = \frac{w^\star - w_{a_{ij}}}{w^\star} \;\forall a_{ij} \tag{31}$$

$$P(r=1|l_{ij}^m = 0, a_{ij}) = 1 \;\forall a_{ij} \tag{32}$$

where $w^\star = w_{\max} + 1$; $w_{\max}$ denotes the maximum edge weight for any edge and any action.

Theorem 3. *Let the CPT of binary variable r in the mixture model be set as per* (31) *and* (32)*, parameters $\boldsymbol{x} = \{\boldsymbol{x}^m \forall m\}$ satisfy the flow constraints* (28)*, and parameters $\boldsymbol{y}$ satisfy* *the budget constraint* (27)*, then maximizing the likelihood* $P(r = 1; \boldsymbol{x}, \boldsymbol{y})$ *of observing* $r = 1$ *in the mixture model is equivalent to solving the relaxed QP in table 1.*

We omit the proof as it is similar to the proof of Thm. 1. The expected log-likelihood $Q(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ that EM maximizes is given below (proof provided in appendix):

$$\propto \sum_{(i,j)} \Big[\sum_{a_{ij}} \log y^\star_{a_{ij}} \Big(\sum_m y_{a_{ij}} \tilde{x}^m_{ij} + \sum_m \hat{w}_{a_{ij}} y_{a_{ij}} x^m_{ij} \Big) \Big] + \sum_m \underbrace{\sum_{(i,j)} \Big[\tilde{x}^m_{ij} \log \tilde{x}^{\star m}_{ij} + x^m_{ij} \big(\sum_{a_{ij}} \hat{w}_{a_{ij}} y_{a_{ij}} \big) \log x^{\star m}_{ij} \Big]} \tag{33}$$

Notice the similarity of the expression under brace in the above equation with that of the expected log-likelihood (16) for the SPR problem. If we replace $\hat{w}_{ij}$ in (16) by $\big(\sum_{a_{ij}} \hat{w}_{a_{ij}} y_{a_{ij}} \big)$, then we can *independently* maximize the expression under brace for each od pair m by using algorithm 1, thereby also increasing the scalability w.r.t. the total number of od pairs. This is possible as the flow constraints are independent for each variable set $\boldsymbol{x}^m$. Thus, the SPR message-passing becomes a subroutine to solve our RNDP problem. The only remaining thing is to maximize the first expression in $Q(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ w.r.t. $\boldsymbol{y}^\star$ and the budget constraint (27). The steps to maximize it are shown in appendix.

Extracting Integral Solution Our goal is to compute an integral $\boldsymbol{y}$ upon convergence of the EM algorithm. Solving the relaxed QP for the RNDP often results in fractional decision $\boldsymbol{y}$. To avoid ad-hoc rounding of the fractional solution, we present an optimization based approach that encourages integral solutions. Observe that for any integral decision $\boldsymbol{y}$, its entropy $-\sum_{a_{ij}} y_{a_{ij}} \ln y_{a_{ij}}$ must be zero. For any fractional $y_{a_{ij}}$, we are guaranteed that the entropy must be positive. We exploit this fact while maximizing $Q(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ w.r.t. $\boldsymbol{y}^\star$ by changing the objective as:

$$\min_{\boldsymbol{y}^\star} - \sum_{(i,j)} \sum_{a_{ij}} \log y^\star_{a_{ij}} \delta_{a_{ij}} - \sum_{(i,j)} \rho_{ij} \sum_{a_{ij}} y^\star_{a_{ij}} \ln y^\star_{a_{ij}} \tag{34}$$

where $\delta_{a_{ij}} = \sum_m y_{a_{ij}} \tilde{x}^m_{ij} + \sum_m \hat{w}_{a_{ij}} y_{a_{ij}} x^m_{ij}$ is the constant term in (33). We also changed the sign of the objective to negative. The penalty weight $\rho_{ij} > 0$ encourages deterministic solutions as their entropy is lower. The above optimization problem is an instance of difference-of-convex functions (DC) programming. The objective function is a different of two convex functions $-\sum_{(i,j)} \sum_{a_{ij}} \log y^\star_{a_{ij}} \delta_{a_{ij}}$ and $\sum_{(i,j)} \rho_{ij} \sum_{a_{ij}} y^\star_{a_{ij}} \ln y^\star_{a_{ij}}$. It can be solved using the concave-convex procedure (CCCP) described in (Yuille and Rangarajan 2001). We show details in the appendix. The resulting approach nicely integrates with EM as it can also be implemented using message-passing. We highlight that this approach is fairly general and also applicable to other types of SPDM problems.

5 Experiments

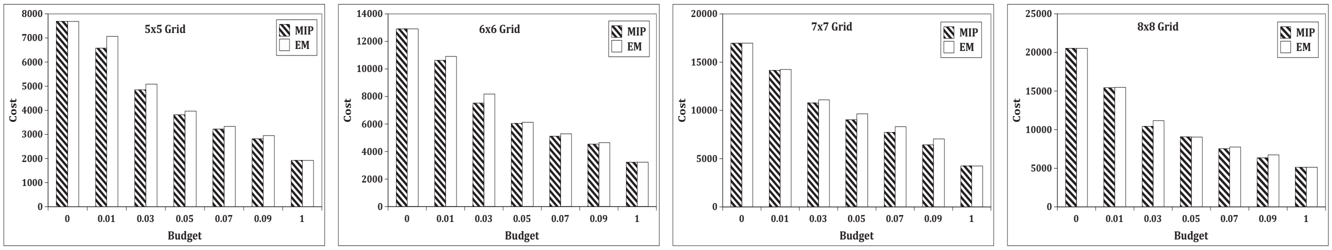
We present results comparing EM against the MIP solver Cplex v12.6 for the RNDP problem. We used grid shaped graphs to simulate realistic road networks, with sizes ranging from 5×5 grid to 20×20 grid. Smallest 5×5 graph has 80 directed edges, and 20×20 graph has 1520 edges. Each edge has three repair action. Action 0 is the default (or *noop*) with zero cost, action 1 has cost randomly chosen between $[40, 400]$ and action 2 has cost twice that of action one's cost. Intuitively, higher cost action leads to lower edge weight. The default weight w_{a_0} of an edge (corresponding to default action a_0) is chosen randomly between $[60, 600]$. The edge weight for action 1 is set as $w_{a_0}/2$ and the edge weight for action 2 is $w_{a_0}/4$ to simulate the higher quality of an expensive repair action. All our experiments are performed on a 16 core linux machine (with 32 parallel threads). Both EM and Cplex were allowed to use 20 parallel threads with 10GB RAM limit. The time cutoff was 3 hours for each algorithm per instance.

Figures 3 and 4 show the quality comparisons between EM and Cplex for a range of budgets. Let $B^{\max}$ denote the budget just sufficient to take the most expensive repair action 2 for each edge in the network. In figures 3(a)-(d), x-axis denotes the fraction of $B^{\max}$ that is allotted ('0.01' means budget for the instance is $0.01 \cdot B^{\max}$). For reference, '0' implies zero budget, and '1' implies full budget. For each grid size $n \times n$, we generated n pairs of origin, destination nodes randomly. To make the problem challenging which would require repairing a large number of roads and also, careful sharing of road segments among shortest paths for different o-d pairs, the origin and destination nodes always lie on the opposing boundaries of the grid. Each data point is an average over 5 randomly generated instances.

From figures 3 and 4 it is quite clear that Cplex provides competitive results with EM only for grid sizes ranging from 5×5 to 10×10. EM's solution is only marginally worse ($<$ 10% additional cost) than Cplex's quality (which was near-optimal for most instances) for these moderate size graphs. These results show that EM was able to provide near-optimal solutions for these problems despite solving a non-convex problem.

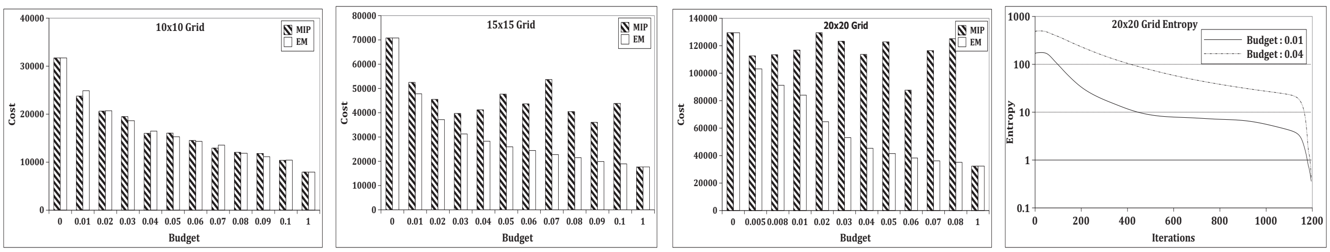
Figures 4(b) and (c) show that for the larger instances, EM significantly outperforms Cplex over a range of budget settings, sometimes providing cost savings as high as 70% for 20×20 grid and '0.08' budget setting. The main reason for Cplex's degraded performance is that due to large problem size (number of variables and constraints) for 15×15 and 20×20 grids, the branch-and-bound strategy of Cplex is unable to explore sufficient number of nodes within the three hour time limit. EM, on-an-average, converges within 1 hour for 15×15 grid and within 5000 seconds for 20×20 grid.

Figure 4(d) shows the efficiency of our entropy-based penalty approach (y-axis in log-scale). We used a fixed penalty weight $\rho = 0.005$ for each edge and started applying penalty from iteration 1000 onwards. In figure 4(d) we show how the total entropy of the network, $\sum_{(i,j)} \sum_{a_{ij}} y_{a_{ij}} \ln y_{a_{ij}}$, evolves with increasing EM iterations for two budget settings ('0.01' and '0.04'). For both these settings, EM steadily decreases the entropy (which is beneficial to extract a deterministic solution) until iteration 1000. The entropy for setting '0.01' is lower than '0.04' as in the former, higher number of repair actions have close to



(a) 5×5 grid (b) 6×6 grid (c) 7×7 grid (d) 8×8 grid

Figure 3: Quality comparisons between EM and MIP solver Cplex for a range of budget and problem sizes. Lower cost is better.



(a) 10×10 grid (b) 15×15 grid (c) 20×20 grid (d) Decrease in entropy with iters.

Figure 4: Quality comparisons between EM and MIP solver Cplex for a range of budget and problem sizes. Lower cost is better.

zero probability due to tighter budget. When entropy-based penalty kicks in at iteration 1000, we see that within the next 200 iterations, the entropy gradually goes to zero, which then permits us to extract a deterministic solution. For '0.04' budget setting, the entropy goes down from 27.4 at iteration 1000 to 0.42 at iteration 1200, showing the significant impact of our approach to get good quality deterministic solutions. Indeed, for every instance in figures 4(a)-(c), we were able to recover an integral solution using our entropy based method. This supports the application of EM and the LM framework to settings where an integral solution is desired.

6 Conclusion

In our work, we have presented a new probabilistic inference and graphical models based perspective on the SPR problem. We have shown how the likelihood maximization (LM) framework and associated solution approaches such as the EM algorithm can be applied to shortest path based decision making (SPDM) problems. The main benefit of such probabilistic viewpoint lies in its ability to generalize to SPDM problems that may be nonlinear, nonconvex and in general, NP-Hard. We addressed one such road network design problem. Empirically, our LM and EM based approach significantly outperformed the standard MIP solver w.r.t. solution quality. Thus, our work introduced a promising new framework which combines SPDM with probabilistic inference, and opens the door to the application of rich inference and optimization-based techniques to solve SPDM problems.

Acknowledgments

Support for this work was provided by the research center at the School of Information Systems at the Singapore Management University.

References

Aksu, D. T., and Ozdamar, L. 2014. A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation. *Transportation Research Part E: Logistics and Transportation Review* 61:56 – 67.

Bertsekas, D. P. 1999. *Nonlinear Programming*. Cambridge, MA, USA: Athena Scientific.

Brent, R. P. 1971. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal* 14(4):422–425.

Bunn, A. G.; Urban, D. L.; and Keitt, T. H. 2012. Landscape connectivity: A conservation application of graph theory. *Journal of Environmental Management* 22:87 – 103.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2001. *Introduction to Algorithms 2nd edition*. MIT Press.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical society, Series B* 39(1):1–38.

Dijkstra, E. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271.

Duque, P. M., and Sorensen, K. 2011. A GRASP metaheuristic to improve accessibility after a disaster. *OR Spectrum* 33(3):525–542.

Ghosh, S.; Kumar, A.; and Varakantham, P. 2015. Probabilistic inference based message-passing for resource constrained DCOPs. In *IJCAI*, 411–417.

Kumar, A., and Zilberstein, S. 2010. Anytime planning for decentralized pomdps using expectation maximization. In *UAI*, 294–301.

Kumar, A.; Zilberstein, S.; and Toussaint, M. 2015. Probabilistic inference techniques for scalable multiagent decision making. *JAIR* 53:223–270.

Liberatore, F.; Ortuno, M.; Tirado, G.; Vitoriano, B.; and Scaparra, M. 2014. A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in humanitarian logistics. *Computers and OR* 42:3–13.

Liu, Q., and Ihler, A. T. 2013. Variational algorithms for marginal MAP. *JMLR* 14(1):3165–3200.

Minor, E. S., and Urban, D. L. 2008. A graph-theory framework for evaluating landscape connectivity and conservation planning. *Conservation Biology* 22:297 – 307.

Ozdamar, L.; Aksu, D. T.; and Ergunes, B. 2014. Coordinating debris cleanup operations in post disaster road networks. *Socio-Economic Planning Sciences* 48(4):249 – 262.

Pallottino, S., and Scutella, M. G. 1998. Shortest path algorithms in transportation models: Classical and innovative aspects. In *Equilibrium and Advanced Transportation Modelling*, Centre for Research on Transportation. Springer US. 245–281.

Pioro, M.; Szentesi, A.; Harmatos, J.; Juttner, A.; Gajowniczek, P.; and Kozdrowski, S. 2002. On open shortest path first related network optimisation problems. *Perf. Eval.* 48(1–4):201 – 223.

Sontag, D., and Jaakkola, T. 2007. New outer bounds on the marginal polytope. In *NIPS*, 1393–1400.

Sontag, D.; Meltzer, T.; Globerson, A.; Jaakkola, T.; and Weiss, Y. 2008. Tightening LP relaxations for MAP using message passing. In *UAI*, 503–510.

Toussaint, M., and Storkey, A. J. 2006. Probabilistic inference for solving discrete and continuous state markov decision processes. In *ICML*, 945–952.

Toussaint, M.; Charlin, L.; and Poupart, P. 2008. Hierarchical POMDP controller optimization by likelihood maximization. In *UAI*, 562–570.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundation and Trends in Machine Learning* 1(1-2):1–305.

Wu, C. F. J. 1983. On the convergence properties of the EM algorithm. *Annals Of Statistics.* 11(1):95–103.

Yanover, C.; Meltzer, T.; Weiss, Y.; Bennett, P.; and Parrado-hernndez, E. 2006. Linear programming relaxations and belief propagation—an empirical study. *JMLR* 7:2006.

Yuille, A. L., and Rangarajan, A. 2001. The concave-convex procedure (CCCP). In *NIPS*, 1033–1040.