We thank all reviewers for their comments and appreciate the fact that all of you carefully read the paper (this is evident from your comments). We will address Reviewers 1, 2, 3 and 4 as **R1**, **R2**, **R3** and **R4** respectively.

**1. Sparsity.** (We say that a graph is sparse for CMPE if it has a bounded (small) $k$-separator for a given $k$). We will make two points in order to alleviate **R1**'s concern that "the proposed method is practically limited because it requires sparse graphs." First, notice that even though the UAI 2010 and 2014 instances have relatively small treewidth (roughly 15-50), their $k$-separator size can be quite large. In other words, the UAI instances are not sparse from the point of view of the CMPE problem and despite this our proposed method works relatively well (primarily because of the MCKP formulation). We will report the $k$-separator sizes in the paper as well as supplement and add a proposition about the relationship between $k$-separator and treewidth in the paper (**R1** and **R2**). Second (in future work), one can develop structure learning algorithms that induce graphical models having small $k$-separators from data, namely use the size of the $k$-separator as inductive bias (inspired by work in the tractable probabilistic models community where an upper bound on complexity of posterior marginal inference is used as inductive bias).

**2. Gurobi and MILP Encoding.** (**R1** and **R4**) We did not use Gurobi because of precision problems. We tried to play with the tolerances provided on the Gurobi website as well as scaling (see Gurobi manual). However, we found that the solutions Gurobi returned were often inferior to SCIP because tolerances/gaps in SCIP can be set to a much smaller value. To alleviate **R4**'s concerns, we will describe our MILP encoding in the extended version of the paper. C++ code for the encoding is already included in the supplementary material.

**3. Significance of CMPE and Experimental Results.** We agree with **R3**'s assessment that a more compelling case can be made with experiments on a concrete real world application. This is part of our future work. However, we believe that we have performed a systematic experimental study on *realistic sized probabilistic models* (used in past UAI competitions) as well as *hard problems* for our proposed method. All reviewers have rightly pointed out the significance of CMPE as a *unifying* query type because many reasoning queries in graphical models can be reduced to it.

**4. $k$-separators.** As far as we know, this is the first paper that uses the concept of $k$-separator for efficient inference in graphical models. It is related to a previously proposed concept called $w$-cutset, but not the same as the latter (**R3**).

**5. Background on MCKP.** (**R4**) An excellent reference for MCKP is the book on Knapsack problems by Kellerer et al. [1]. We have included multiple references for knapsack solvers in the paper (including the book above). However, in order to make the paper self contained, we will try to describe the specific solver used in more detail. Thank you for the suggestion. The good news is that (and as mentioned in the paper) we can leverage advances in MCKP solvers to improve the efficiency and scalability of CMPE solvers because of our proposed method.

**6. Relationship to Mixed Networks.** Notice that there is just one global constraint in CMPE. Therefore we did not use the mixed networks framework. The latter is useful when you have a number of local constraints defined over a subset of variables and efficient constraint propagation techniques (e.g., arc consistency, path consistency, etc.) exist for handling the local constraints. Moreover, in presence of local constraints, the problem can be solved (exactly) in time and space that scales exponentially with the treewidth of the combined primal graph. CMPE is a much harder task and remains NP-hard even on bounded treewidth combined primal graphs. (**R4**)

**7. Using MPE solvers for CMPE.** This approach will be very inefficient. The constraint in CMPE will cause minimal pruning and the search procedure will enumerate a large number of assignments. It will be only useful for assignments that are at or very near the (unconstrained) MPE value. Again, we want to emphasize that CMPE is much harder than MPE, both for bounding and solving. For example, the MPE solver approach will be inefficient even if the combined primal graph is empty. On empty graphs MPE can be solved in linear time while CMPE remains NP-hard. (**R4**).

**8. Other Minor Points Raised by the Reviewers.**

- Proposition 1 is a claim. It says that if you find two assignments $\mathbf{x}^u$ and $\mathbf{x}^l$ by solving the CMPEs defined in the equation above the proposition, then the nearest assignment is either $\mathbf{x}^u$ or $\mathbf{x}^l$. Thus, NAP can be solved if you solve CMPE. (**R1**)
- We agree that using "volume" instead of "cost" makes more sense. However, the term "cost" is often used in the Knapsack literature and we were just being consistent. (**R1**)

## References


[1] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Germany, 2004.