Erratum to “An algorithm for ranking assignments using reoptimization” [Computers and Operations Research 35 (2008) 3714-3726

Christian Roed Pedersen

Department of Operations Research, Aarhus University, Ny Munkegade, Building 1530, DK-8000 Aarhus C, Denmark.

Lars Relund Nielsen

Department of Genetics and Biotechnology, Aarhus University, P.O. Box 50, DK-8830 Tjele, Denmark.

Kim Allan Andersen*

Department of Business Studies, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark.

April 16, 2009

In the paper Pedersen, Nielsen, and Andersen [5] we developed an algorithm for ranking $n \times n$ assignments using reoptimization and compare our algorithm with other algorithms with the same time complexity. However, as pointed out by Dr. A. Volgenant, we unfortunately missed one available implementation written by Miller, Stone, and Cox [3] in *IEEE Transactions on Aerospace and Electronic Systems*.

The algorithm of Miller et al. [3] (`Miller`) and Pedersen et al. [5] (`DU1`) are both based on the branching technique presented in Murty [4] where the set of possible assignments is partitioned into at most $n-1$ disjoint subsets for each additional ranking made. In each subset the best assignment is found by applying the successive shortest path procedure implementation of Jonker and Volgenant [2]. The algorithms differ in the following ways:

1. In `DU1` dual variables are updated before reoptimization. This is not the case in `Miller`.
2. In `Miller` (partial) assignments are saved while maintaining the candidate set. In `DU1` the optimal assignment is recalculated each time a subset is selected from the candidate set to reduce memory size.
3. An interval heap is used in `DU1` to maintain a priority deque of the candidate set and keeping the size of the queue low. `Miller` uses a priority queue of the candidate set.
4. In `Miller` an assignment is selected from the candidate set based on a lower bound. This is not the case in `DU1`.
5. `Miller` uses a heuristic to select subsets from the candidate set with many fixed variables.
6. `DU1` uses a heuristic that improve the way two subsets with the same optimal value are inserted into the candidate set.

From a theoretical point of view items 1 and 6 will speed up `DU1` compared to `Miller` while items 2-5 will speed up `Miller` compared to `DU1`. However, due to items 2 and 3, `Miller` requires more memory. Moreover, the heuristic must be fast and the lower bound tight to speed up `Miller`.

We obtained the source code of `Miller`, written in C, from Professor I.J. Cox's homepage[1], with his permission. The algorithms were compared using the Beasley test instances [1]. The tests were performed on a Dualcore Intel EMT64 Xeon 2.80 GHz with 1 MB cache and 6 GB RAM using Ubuntu linux. The results, averaged over five runs, are presented in Figure 1.

Figure 1 shows that `Miller` has a better performance than `DU1` for the smaller instances $n \in \{100, 200, 300\}$. For the instances $n \in \{400, 500, 600, 700\}$ neither of the algorithms outperform the

*Corresponding author, e-mail: kia@asb.dk.

[1] http://www.adastral.ucl.ac.uk/∼ icox/

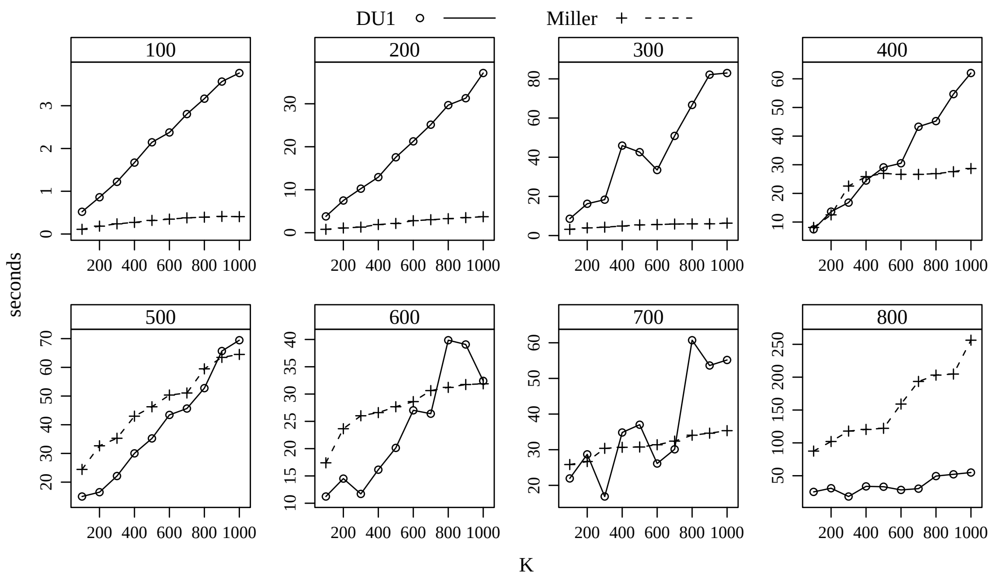

Figure 1: CPU times for `Miller` and `DU1`.

other. For the instance $n = 800$ `DU1` performs better. It appears that the results obtained by `DU1` are more unstable than the ones obtained by `Miller` which is due to the candidate set implementation shortly described in item 3 above.

Acknowledgement

We thank Dr. A. Volgenant for directing our attention to the paper by Miller et al. [3].

References

[1] J.E. Beasley. Linear programming on cray supercomputers. *The Journal of the Operational Research Society*, 41(2):133–139, 1990.

[2] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[3] M.L. Miller, H.S. Stone, and I.J. Cox. Optimizing murty's ranked assignment method. *IEEE Transactions on Aerospace and Electronic Systems*, 33:851–862, 1997.

[4] K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687, 1968.

[5] C.R. Pedersen, L.R. Nielsen, and K.A. Andersen. An algorithm for ranking assignments using reoptimization. *Computers & Operations Research*, 35(11):3714–3726, 2008.