As usual with pure Hungarian methods, ASSCT turns out to be very sensitive to cost range. Only for small cost ranges the use of pointer techniques makes the algorithm competitive. LSAP performs strangely on the cost range $1-100$ with relatively large computation times, also observed by Derigs and Metz [9]. LAPJV is clearly faster than ASSIGN, and less sensitive to the range of the cost coefficients.

For a comparison on sparse problems we adapted the data structure of LAPJV, yielding LAPJVsp. Average computation times (for ten problems) are compared in Table 4 with those of two algorithms for sparse LAPs:

- SPASS: Lawler's $O(n^3)$ Hungarian method [23] coded by Carpaneto and Toth [6],
- ASSIGN: again Bertsekas' algorithm [3] as provided to us (ASSIGN requires too much memory for LAPs with $n=400$ and 20% density of the costs matrix).

An indirect comparison may be made with the code SPTM from Carraresi and Sodini [7] for sparse LAPs. The SPTM computation times in Table 4 have been obtained by multiplying the SPASS times with the ratios calculated from [7]. ASSIGN and LAPJVsp clearly outperform the Hungarian method. The margin in speed of LAPJVsp over ASSIGN is about the same as on full density problems.

Table 4. *Computation times for sparse assignment problems (in ms) ("." indicates not run or not known)*

density and cost range	problem size	LAP algorithm SPASS	ASSIGN	SPTM	LAPJVsp
5%/1 – 100	100	65	38		19
	200	211	110	67	62
	400	713	451	335	253
5%/1 – 1000	100	82	50		25
	200	361	174	113	81
	400	1553	688	356	333
20%/1 – 100	100	71	54		26
	200	304	290	234	154
	400	1046		1029	657
20%/1 – 1000	100	119	69		33
	200	576	384	253	188
	400	2123		1039	788

We may also compare results with the shortest augmenting paths algorithms for sparse LAPs of Derigs and Metz [8]. Their best code SAPM3 is faster than SPASS: 40% to 50% on problems with $n=200$, about 5% density, and on cost ranges $1-100$ and $1-1000$. This is substantially slower than LAPJVsp. Consequently, the code LAPJVsp will provide a better basis for an in-core-out-of-core algorithm for full density LAPs as proposed by Derigs and Metz [9]. Such a code solves a sparse version of full density problems that cannot be entirely contained in memory. It contains a procedure to check whether the not considered assignments price out correctly, and a reoptimization procedure for use if they do not. An in-core-out-of-core code can also be used as an all in-core code. The advantages of solving only sparse problems has to be weighed against the effort to construct the sparse problems. Unfortunately this task is computationally non-trivial, leading to about the same total computation times as for LAPJV.

9. Conclusions

The computational results show that the average computation times of the algorithm LAPJV are uniformly lower than the best of other algorithms. The code is of moderate size, and the memory requirements are small. The algorithm is suited for both dense and sparse assignment problems, and its sensitivity to cost range is relatively low.

References

[1] Balinski, M. L.: Signature methods for the assignment problem. Operations Research *33*, 527 – 536 (1985).

[2] Barr, R., Glover, F., Klingman, D.: The alternating path basis algorithm for assignment problems. Mathematical Programming *13*, 1 – 13 (1977).

[3] Bertsekas, D. P.: A new algorithm for the linear assignment problem. Mathematical Programming *21*, 152 – 171 (1981).

[4] Burkard, R. E., Derigs, U.: Assignment and Matching Problems: Solution Methods with Fortran Programs, pp. 1 – 11. Berlin-Heidelberg-New York: Springer 1980.

[5] Carpaneto, G., Toth, P.: Algorithm 548 (solution of the assignment problem). ACM Transactions on Mathematical Software *6*, 104 – 111 (1980).

[6] Carpaneto, G., Toth, P.: Algorithm 50: algorithm for the solution of the assignment problem for sparse matrices. Computing *31*, 83 – 94 (1983).

[7] Carraresi, P., Sodini, C.: An efficient algorithm for the bipartite matching problem. European Journal of Operational Research *23*, 86 – 93 (1986).

[8] Derigs, U., Metz, A.: An efficient labeling technique for solving sparse assignment problems. Computing *36*, 301 – 311 (1986).

[9] Derigs, U., Metz, A.: An in-core/out-of-core method for solving large scale assignment problems. Zeitschrift für Operations Research *30*, 181 – 195 (1986).

[10] Dorhout, B.: Het Lineaire Toewijzingsprobleem: Vergelijking van Algorithmen. Rapport BN 21/73, Mathematisch Centrum, Amsterdam (1973).

[11] Dorhout, B.: Experiments with some algorithms for the linear assignment problem. Report BW 39, Mathematisch Centrum, Amsterdam (1975).

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

[13] Edmonds, J., Karp, R. M.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM *19*, 248 – 264 (1972).

[14] Ford jr., L. R., Fulkerson, D. R.: Flows in Networks. Princeton: Princeton University Press 1962.

[15] Glover, F., Klingman, D., Phillips, N.: A new polynomially bounded shortest path algorithm. Operations Research *33*, 65 – 73 (1985).

[16] Glover, F., Klingman, D., Phillips, N., Schneider, R.: New polynomial shortest path algorithms and their computational attributes. Management Science *31*, 1106 – 1128 (1985).

[17] Goldfarb, D.: Efficient dual simplex algorithms for the assignment problem. Mathematical Programming *33*, 187 – 203 (1985).

[18] Hung, M. S., Rom, W. O.: Solving the assignment problem by relaxation. Operations Research *28*, 969 – 982 (1980).

[19] Jonker, R.: Traveling salesman and assignment algorithms: design and implementation. Faculty of Actuarial Science and Econometrics, University of Amsterdam (1986).