

instance	Quad-tree		Support-based	
	mean	st. dev.	mean	st. dev.
500A	635461.0	4953.8	758585.8	6567.3
500B	548185.3	6864.1	653769.3	7307.3
500C	531152.3	7631.5	642077.7	8158.2
500D	593904.0	4658.0	712250.0	5818.4

Table 1: Number of search nodes explored within a time limit of 30 seconds on instances of the bi-objective binary knapsack problem with 500 items.

instance	Quad-tree		Quad-tree (bal.)		Support-based	
	mean	st. dev.	mean	st. dev.	mean	st. dev.
KroAB100	505.3	17.6	60034.0	2565.0	63850.0	227.3
KroAB150	234.3	9.4	23776.3	950.1	25138.7	339.8
KroAB200	361.0	1.6	18574.0	542.0	21443.3	143.5
KroAB300	247.3	17.9	9080.3	362.4	9337.7	176.4
KroAB400	176.0	0.4	4516.0	174.5	4798.3	70.4
KroAB500	116.0	1.6	3893.7	137.7	4033.0	107.3

Table 2: Number of search nodes explored within a time limit of 30 seconds on instances of the bi-objective traveling salesman problem with 100, 150, 200, 300, 400, and 500 cities.

unknown and the given set is the union of the approximation of many state-of-the-art algorithm to solve this problem (Lust and Teghem 2010). Since the internal structure of the Pareto constraint is initialized in advance, we consider both cases of an unbalanced and well-balanced quad-tree. As for the previous experiments, the mean and standard deviation over 10 runs are presented in the Table 2.

Again, the support-based is the fastest approach. As mentioned above, we observe that the quad-tree algorithm is very sensitive to the order in which solutions are added in its structure. On one hand, a well-balanced quad-tree is competitive with the support-based algorithm (but still always dominated). On the other hand, a poorly balanced quad-tree substantially deteriorates the number of search nodes explored. Interestingly, the balanced quad-tree and the support-based algorithms were able to add 3 solutions in the approximation of the KroAB300 efficient set and 4 solutions in the approximation of the KroAB500 efficient set. This experiment illustrates the flexibility of the Pareto constraint and one of its possible uses to improve an existing approximation of the efficient set or to prove its optimality.

Conclusion

This paper introduced a support-based algorithm to implement the bi-objective Pareto constraint. This incremental filtering algorithm relies on the bi-objective ordering property of BOCO problems. Experiments demonstrate that the support-based algorithm is more efficient than the classical algorithm used to implement the Pareto constraint. Also, this algorithm is simpler to implement than the quad-tree based propagator since it only relies on a linked list and two reversible pointers.

References

- Ehrgott, M. 2005. *Multicriteria optimization*, volume 2. Springer Berlin.
- Finkel, R. A., and Bentley, J. L. 1974. Quad trees a data structure for retrieval on composite keys. *Acta informatica* 4(1):1–9.
- Gavannelli, M. 2002. An algorithm for multi-criteria optimization in CSPs. *ECAI'02* 136–140.
- Habenicht, W. 1983. Quad trees, a datastructure for discrete vector optimization problems. In *Essays and Surveys on Multiple Criteria Decision Making*. Springer. 136–145.
- Haimes, Y. Y.; Lasdon, L. S.; and Wismer, D. A. 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics* 1(3):296–297.
- Le Pape, C.; Couronné, P.; Vergamini, D.; and Gosselin, V. 1994. Time-versus-capacity compromises in project scheduling.
- Lust, T., and Teghem, J. 2010. Two-phase Pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics* 16(3):475–510.
- Mavrotas, G. 2007. Generation of efficient solutions in multiobjective mathematical programming problems using GAMS. Effective implementation of the ϵ -constraint method.
- Oscar Team. 2012. OscarR: Scala in OR. Available from <https://bitbucket.org/oscarlib/oscar>.
- Paquete, L., and Stützle, T. 2009. Design and analysis of stochastic local search for the multiobjective traveling salesman problem. *Computers & operations research* 36(9):2619–2631.
- Ralphs, T. K.; Saltzman, M. J.; and Wiecek, M. M. 2006. An improved algorithm for solving biobjective integer programs. *Annals of Operations Research* 147(1):43–70.
- Samet, H. 2006. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.
- Schaus, P., and Hartert, R. 2013. Multi-Objective Large Neighborhood Search. In *19th International Conference on Principles and Practice of Constraint Programming*.
- Van Wassenhove, L. N., and Gelders, L. F. 1980. Solving a bicriterion scheduling problem. *European Journal of Operations Research* 4:42–48.
- Xavier Gandibleux. 2013. A collection of test instances for multiobjective combinatorial optimization problems. Available from <http://xgandibleux.free.fr/MOCOLib/>.
- Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C. M.; and da Fonseca, V. G. 2003. Performance assessment of multi-objective optimizers: An analysis and review. *Evolutionary Computation, IEEE Transactions on* 7(2):117–132.