Subset Approximation of Pareto Regions with Bi-objective A* (Extended Abstract)*

Jorge A. Baier[1,2], **Nicolás Rivera**[3], **Carlos Hernández Ulloa**[4]
[1]Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Santiago, Chile
[2]Instituto Milenio Fundamentos de los Datos, Santiago, Chile
[3]Instituto de Ingeniería Matemática, Universidad de Valparaíso, Valparaíso, Chile
[4]Facultad de Ingeniería y Tecnología, Universidad San Sebastián, Bellavista 7, 84205254, Santiago, Chile
jabaier@ing.puc.cl, n.a.rivera.aburto@gmail.com, carlos.hernandez@uss.cl.

1 Introduction

In bi-objective search we are given a graph G in which each arc, and thus each path, is associated with a pair of non-negative costs, which represent meaningful objective functions. For example, in transportation, one function could refer to the time required to traverse an edge while the other could refer to fuel consumption. To compare two paths, a *dominance* relation is used. Path π_1 dominates path π_2 if both components of the cost of π_1 are less than or equal to the respective components of the cost of π_2 and their costs are not equal. Given a start vertex and a goal vertex in G, the problem consists of finding a Pareto-optimal solution set which contains all paths from start to goal which are not dominated by another path from start to goal.

Bi-objective search is required for several real-world applications; notably in transportation and logistics when time and cost (e.g., fare) are minimized (e.g., Pallottino and Scutella 1998; Bronfman et al. 2015; Müller-Hannemann and Weihe 2006), or when time and risk are minimized for cycling (Ehrgott et al. 2012). Recently, it has also been used in AI problems like robot planning (Davoodi 2017) and multi-agent path finding (Ren, Rathinam, and Choset 2021).

An important hurdle to bi-objective search is the size of the solution set, which can be exponential on the size of the graph (Hansen 1980). As a consequence, bi-objective search algorithms may only compute a handful of solutions before running out of time. Worse even, because of the exhaustive nature of their search, such solutions may not represent the diversity of the solution set. To address this problem, approaches to approximating the solution set have been proposed. One line of work proposes algorithms that reduce high runtimes by computing a solution set with approximate solutions whose suboptimality is bounded (e.g., Warburton 1987; Perny and Spanjaard 2008; Goldin and Salzman 2021). Another less explored line of work computes *subset approximations* (e.g., Cohon 1978; Henig 1986), in which a subset of the solution set is computed. A limitation of the former approach is that even though approximate solutions may be faster to compute still a large number of solutions may have to be computed. The main limitation of the latter approach is that the maximum number of computable solutions is fixed and task-dependent. This does not allow returning more solutions if more search time is available.

2 Notation

Boldface lower case letters indicate column vectors in $\mathbb{R}^2$. The first and second component of $\mathbf{p}$ are denoted by $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively. We consider standard addition and multiplication by scalar of vectors. We say that $\mathbf{p} \preceq \mathbf{q}$, iff $\mathbf{p}_1 \leq \mathbf{q}_1$ and $\mathbf{p}_2 \leq \mathbf{q}_2$; in addition, $\mathbf{p} \prec \mathbf{q}$ iff $\mathbf{p} \preceq \mathbf{q}$ and $\mathbf{p} \neq \mathbf{q}$. We say that $\mathbf{p}$ *dominates* $\mathbf{q}$ when $\mathbf{p} \prec \mathbf{q}$, and that $\mathbf{p}$ *weakly dominates* $\mathbf{q}$ when $\mathbf{p} \preceq \mathbf{q}$

A *path* π from s_1 to s_n on a graph $G = (S, E)$ is a sequence of states $s_1, s_2, \ldots, s_n$ such that $(s_i, s_{i+1}) \in E$ for all $i \in \{1, \ldots, n-1\}$. Given a path $\pi = s_1, \ldots, s_n$, its cost is given by $\sum_{i=1}^{n-1} \mathbf{c}(s_i, s_{i+1})$ and denoted by $\mathbf{c}(\pi)$. Path π dominates path π' if and only if $\mathbf{c}(\pi) \prec \mathbf{c}(\pi')$.

A *bi-objective search instance* is as a tuple $(S, E, \mathbf{c}, s_0, s_g)$, where (S, E) is a graph, s_0 and s_g are, respectively, the *start state* and the *goal state*, and $\mathbf{c} : E \to \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0}$ is a non-negative *cost function*. A *start-to-goal path* is a path from s_0 to s_g. The *Pareto-optimal solution set*, denoted by sols_P, contains all start-to-goal paths that are not dominated by another one.

3 Solution Subsets via Bi-Objective Search

Now we describe our approach to obtain a subset of the Pareto-optimal solution set. The approach can be used along with any bi-objective search algorithm. Our idea is to map the problem P into another search problem $P_{\alpha,\beta}$ where α and β are two parameters that control the precision of our approximation.

We shall assume that $\alpha, \beta \in (0, 1]$ with $\alpha + \beta > 1$. To build $P_{\alpha,\beta}$ we define the matrix $M_{\alpha,\beta}$ given by:

$$M_{\alpha,\beta} = \begin{pmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{pmatrix},$$

and define $P_{\alpha,\beta} = (S, E, \mathbf{c}_{\alpha,\beta}, s_0, s_g)$, where for each $e \in E$, we define $\mathbf{c}_{\alpha,\beta}(e)$ as the result of applying the matrix $M_{\alpha,\beta}$ to $\mathbf{c}(e)$ that is

$$M_{\alpha,\beta}\,(\mathbf{c}(e)) = (\alpha \mathbf{c}_1(e) + (1-\alpha)\mathbf{c}_1(e), (1-\beta)\mathbf{c}_1 + \beta \mathbf{c}_2(e)).$$

This new instance $P_{\alpha,\beta}$ has two important properties. The first property is that it defines a dominance relation $\preceq_{\alpha,\beta}$ in

*Originally published in the Proceedings of AAAI-22 which $\mathbf{u}$ is dominated by $\mathbf{v}$ if and only if $\alpha\mathbf{u}_1+(1-\alpha)\mathbf{u}_2 \leq \alpha\mathbf{v}_1 + (1-\alpha)\mathbf{v}_2$ and $(1-\beta)\mathbf{u}_1 + \beta\mathbf{u}_2 \leq (1-\beta)\mathbf{v}_1 + \beta\mathbf{v}_2$, and $\mathbf{u} \neq \mathbf{v}$, or much shorter, $M_{\alpha,\beta}\mathbf{u} \preceq M_{\alpha,\beta}\mathbf{v}$. The second important property is that the solution set of $P_{\alpha,\beta}$ is contained in the solution set of P.

Theorem 1. *Let $P = (S, E, \mathbf{c}, s_0, s_g)$ be a bi-objective search instance and let $P_{\alpha,\beta}$ be defined as above with $\alpha, \beta \in (0,1]$ and $\alpha + \beta > 1$. Then $\mathrm{sols}_{P_{\alpha,\beta}} \subseteq \mathrm{sols}_P$.*

We can find solutions to the new instance $P_{\alpha,\beta}$ using existing bi-objective search technology. Indeed, if we have a heuristic function $\mathbf{h}$ for the original problem P, we can obtain a heuristic for the new problem by applying $M_{\alpha,\beta}$ to the original heuristic $\mathbf{h}$. Henceforth, we denote by $\mathbf{h}_{\alpha,\beta}$ the result of applying $M_{\alpha,\beta}$ to $\mathbf{h}$.

Our theoretical analysis implies that when using small values of α and β more pruning is performed, and thus we shall expect faster executions. As a consequence, we propose a simple approach leading to an anytime bi-objective search algorithm. The main idea is to solve the target task for an initial pair of values, e.g. $\alpha = \beta = 0.8$, to then increase both parameters, and repeat until we reach $\alpha = \beta = 1$.

4 Experimental Evaluation

Our evaluation had the objective of evaluating the performance of BOA* (Hernández et al. 2020), a state-of-the-art bi-objective search algorithm, run over $P_{\alpha,\beta}$ with different (α, β).

We evaluated our approach, implemented in C, on maps of the 9th DIMACS Implementation Challenge: Shortest Path[1]; specifically, 50 random instances for each of four USA road maps used by Machuca and Mandow (2012). The cost components represent travel distances (c_1) and times (c_2). The heuristic $\mathbf{h}$ corresponds to the exact travel distances and times to the goal state, computed with Dijkstra's algorithm.

Dividing the Pareto Frontier in Buckets To report the diversity of solutions, imagine that we divide the upper-right quadrant of the plane (i.e. when both coordinates are positive) into five slices by drawing rays starting at the origin forming 18, 36, 54, and 72 degrees with the x-axis. We call each 18-degree slice a "bucket". By counting how many solutions are in each bucket we obtain a measure of diversity.

Table 1 reports our results for 50 random instances for each road map. It reports the total runtime in seconds required to compute the solution set for $P_{\alpha,\beta}$ for each (α, β) pair, and the percentage of solutions that appear in each bucket. In addition, the table reports the total number of solutions in each bucket. We have the following observations:

- We obtain solutions that, on average, are diverse. Indeed the maximum percentage difference between two buckets is equal to 13%.
- 10% of solutions is obtained in about one order of magnitude less time than that required to find all solutions.
- When (α, β) values increase, the runtime increases and the number of solutions found in each bucket increases.

[1] http://users.diag.uniroma1.it/challenge9/download.shtml

		Bucket				
$\alpha = \beta$	$t(s)$	1	2	3	4	5
New York City (NY): 264,346 states, 730,100 edges						
0.80	0.5	15%	19%	18%	18%	21%
0.84	0.7	22%	25%	25%	27%	32%
0.88	1.1	34%	37%	36%	36%	42%
0.92	1.7	53%	54%	57%	60%	65%
0.96	2.5	68%	67%	73%	74%	76%
1	3.9	100%	100%	100%	100%	100%
# solutions:		1,924	2,744	3,370	2,001	1,931
San Francisco Bay (BAY): 321,270 states, 794,830 edges						
0.80	0.5	4%	7%	11%	13%	17%
0.84	0.8	8%	12%	17%	16%	21%
0.88	1.4	16%	20%	26%	22%	26%
0.92	2.4	28%	33%	38%	32%	39%
0.96	4.0	48%	54%	59%	49%	56%
1	8.5	100%	100%	100%	100%	100%
# solutions:		3,509	2,989	3,343	2,230	2,245
Colorado (COL): 435,666 states, 1,042,400 edges						
0.80	1.3	5%	5%	4%	6%	15%
0.84	2.1	9%	10%	6%	8%	18%
0.88	3.7	15%	15%	9%	13%	24%
0.92	7.2	27%	26%	20%	23%	31%
0.96	15.6	45%	44%	36%	41%	45%
1	46.5	100%	100%	100%	100%	100%
# solutions:		8,272	8,140	6,946	4,937	6,431
Florida (FL): 1,070,376 states, 2,712,798 edges.						
0.80	9.7	7%	6%	9%	8%	11%
0.84	16.8	13%	11%	15%	18%	20%
0.88	32.2	22%	20%	26%	32%	31%
0.92	50.8	35%	36%	40%	46%	45%
0.96	85.4	60%	61%	66%	71%	68%
1	210.9	100%	100%	100%	100%	100%
# solutions:		13,019	16,817	17,143	8,689	11,009

Table 1: Results on 50 random instances for different road maps. In all these experiments we set $\alpha = \beta$. The table shows the α (and β), and the total runtime to solve all the instances, and the percentage of solutions in each bucket.

- The relation between computation time and percentage of solutions does not appear to be proportional as one would expect. For example in the FL map we compute approximately 8% of the solutions in about $9.7/210.9 = 4.6\%$ of the time that is needed to compute 100% of solutions. Likewise, to compute around 15% of the solutions we require $16.8/210.9 = 8\%$ of the time required to compute 100% of the solutions.

5 Conclusions

We presented a new approach to subset approximation of the solution set, that can be used as the basis for an anytime bi-objective search algorithm. Our approach transforms the given task into a target bi-objective search task using two real parameters. For each parameter setting, the solutions to the target task is a subset of the solution set of the original task. Depending on the parameters used, the solution set of the target task may be computed very quickly. We prove that our approach is correct and that Bi-Objective A* prunes at least as many nodes when run over the target task.

Acknowledgements

Nicolás Rivera was supported by ANID FONDECYT grant number 3210805. Jorge Baier and Carlos Hernández are grateful to the Centro Nacional de Inteligencia Artificial CENIA, FB210017, BASAL, ANID.

References

Bronfman, A.; Marianov, V.; Paredes-Belmar, G.; and Lüer-Villagra, A. 2015. The maximin HAZMAT routing problem. *European Journal of Operational Research*, 241(1): 15–27.

Cohon, J. L. 1978. Multiobjective Programming and Planning, volume 140 of. *Mathematics in Science and Engineering*.

Davoodi, M. 2017. Bi-objective path planning using deterministic algorithms. *Robotics and Autonomous Systems*, 93: 105–115.

Ehrgott, M.; Wang, J. Y.; Raith, A.; and Van Houtte, C. 2012. A bi-objective cyclist route choice model. *Transportation research part A: policy and practice*, 46(4): 652–663.

Goldin, B.; and Salzman, O. 2021. Approximate Bi-Criteria Search by Efficient Representation of Subsets of the Pareto-Optimal Frontier. In Biundo, S.; Do, M.; Goldman, R.; Katz, M.; Yang, Q.; and Zhuo, H. H., eds., *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS)*, 149–158. AAAI Press.

Hansen, P. 1980. Bicriterion path problems. In *Multiple criteria decision making theory and application*, 109–127. Springer.

Henig, M. I. 1986. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2): 281–291.

Hernández, C.; Yeoh, W.; Baier, J.; Zhang, H.; Suazo, L.; and Koenig, S. 2020. A simple and fast bi-objective search algorithm. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS)*, 143–151.

Machuca, E.; and Mandow, L. 2012. Multiobjective heuristic search in road maps. *Expert Systems with Applications*, 39(7): 6435–6445.

Müller-Hannemann, M.; and Weihe, K. 2006. On the cardinality of the Pareto set in bicriteria shortest path problems. *Annals of Operations Research*, 147(1): 269–286.

Pallottino, S.; and Scutella, M. G. 1998. Shortest path algorithms in transportation models: classical and innovative aspects. In *Equilibrium and advanced transportation modelling*, 245–281. Springer.

Perny, P.; and Spanjaard, O. 2008. Near Admissible Algorithms for Multiobjective Search. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, volume 178, 490–494.

Ren, Z.; Rathinam, S.; and Choset, H. 2021. Multi-objective Conflict-based Search for Multi-agent Path Finding. *CoRR*, abs/2101.03805.

Warburton, A. 1987. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1): 70–79.