Must-Expand Nodes in Multi-Objective Search [Extended Abstract]

Shawn Skyler,[1] Shahaf Shperberg,[1] Dor Atzmon,[2] Ariel Felner,[1] Oren Salzman,[3] Han Zhang,[4] Shao-Hung Chan,[4] Sven Koenig,[4] William Yeoh,[5] Carlos Hernández Ulloa[6]

[1] Ben-Gurion University [2] Royal Holloway, University of London [3] Technion—Israel Institute of Technology
[4] University of Southern California [5] Washington University in St. Louis [6] Universidad San Sebastián
shawn@post.bgu.ac.il, shperbsh@bgu.ac.il, dor.atzmon@rhul.ac.uk, felner@bgu.ac.il, osalzman@cs.technion.ac.il,
zhan645@usc.edu, shaohung@usc.edu, skoenig@usc.edu, wyeoh@wustl.edu, carlos.hernandez@uss.cl

Introduction

In the *Multi-Objective Shortest-Path* (MOSP) problem, each edge in the graph is associated with a *d-dimensional vector* of costs ($\mathbf{c} : E \rightarrow \mathbb{R}^d_{\geq 0}$). **Boldface** font is used to represent d-dimensional vectors. The aim is to find the *Pareto-Optimal Frontier* (POF) of paths between s_{start} and s_{goal} with the best trade-offs between the costs, i.e., a set of *undominated* paths from s_{start} to s_{goal} in which the cost in one dimension cannot be decreased without increasing the cost in other dimensions. Formally, $\mathbf{u}$ *dominates* $\mathbf{v}$ ($\mathbf{u} \prec \mathbf{v}$) if $v_i \leq u_i$, for every $i \in \{1, \ldots, d\}$ and $\mathbf{u} \neq \mathbf{v}$. Path π *dominates* path π' if $\mathbf{c}(\pi) \prec \mathbf{c}(\pi')$. Different *Multi-Objective Search* (MOS) algorithms were developed for solving MOSP (Skriver 2000; Clímaco and Pascoal 2012), among which *best-first* search algorithms only expand nodes with undominated **f**-values (Stewart and White III 1991; Mandow and De La Cruz 2005; Hernandez et al. 2020).

In *Single-Objective Search* (SOS), where $d = 1$, Dechter and Pearl (1985) characterized the set of nodes that any unidirectional search algorithm must expand to prove the optimality of solutions. This theory was extended to bidirectional search algorithms (Eckerle et al. 2017), in which the search is simultaneously performed from both s_{start} and s_{goal}.

In this manuscript, we define for MOS conditions on which nodes must be expanded to prove the optimality of solutions, which nodes should not be expanded (as they cannot lead to a solution), and which nodes may be expanded. In addition, we consider the issue of *Ordering Functions*, which are used by best-first MOS algorithms to decide which node to expand next based on their **f**-values. We present several Ordering Functions and compare them experimentally.

Classification of Nodes

We next generalize common knowledge in classical SOS, to MOS and define different classes of nodes. Fig. 1(Left) illustrates how the **f**-values are mapped to the different areas. The axes correspond to two objectives (i.e., $d = 2$). We assume that admissible **h**-values are used which estimate the distance from the current node to the goal along each of the objective individually.

Never-Expand Nodes (NENs) are dominated by at least one path in POF. NENs are located area D including the dashed lines. Formally, any node n such that $\exists \pi \in$ POF : $\mathbf{c}(\pi) \prec \mathbf{f}(n)$. Analogous to nodes with $f(n) > C^*$ in SOS.

Maybe-Expand Nodes (MBENs) are nodes n, such that $\exists \pi \in$ POF : $\mathbf{f}(n) = \mathbf{c}(\pi)$. All nodes in area C are MBENs. Analogous to nodes with $f(n) = C^*$ in SOS. Area C includes all solutions in the POF.

Must-Expand Nodes (MENs) are all nodes that are not dominated by any solution in the POF. We divide them into two groups: *Domination Nodes* and *Verification Nodes*, their union is the set of MENs. **Domination Nodes** (area A) dominate one solution (or more) from the POF. Formally, node n belongs to area A iff $\exists \pi \in$ POF : $\mathbf{f}(n) \prec \mathbf{c}(\pi)$. Analogous to the MENs in SOS where $f(n) < C^*$. These nodes must be expanded to find all POF. **Verification Nodes** (area B) are undominating and undominated by any path in POF. This is the only type of nodes that does not have an analogy in SOS. These nodes have to be expanded to ensure there are no more solutions in the POF. We note that Mandow and De La Cruz (2005) provided relevant analysis on MENs when proving the correctness of the NAMOA* algorithm.

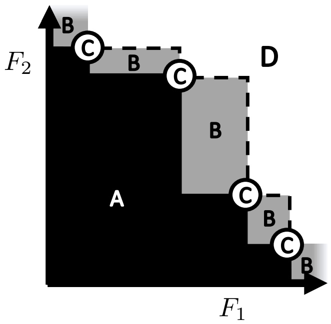
Ordering Functions

An *Ordering Function O* receives two nodes n and m from OPEN and based on their **f**-valuesreturns which node should be expanded before the other, thereby determining the order of the nodes in OPEN. Many Ordering Functions exist; we provide some examples in addition to *Lex* which is commonly used in MOS.

Lexicographical Ordering (*Lex*). *Lex* chooses the node that has a lower value in the (lexicographically) first objective. If there is a tie, it prefers nodes with lower values in the second objective, and so on. Naturally, the $d!$ permutations of the objectives resolve in $d!$ *Lex* orderings.

(Weighted) Average (or sum) Ordering (*Avg*). For nodes n and m and a vector of weights $\mathbf{w}$, *Avg* chooses the node with $\min\left(\sum_{i=1}^{d} w_i \cdot f_i(n)\ ,\ \sum_{i=1}^{d} w_i \cdot f_i(m)\right)$.

Maximum (Minimum) Ordering (*Max*, *Min* resp.). First, *Max* (resp. *Min*) orders the objectives of each node in decreasing (resp. increasing) order. Then, compares the ordered objectives lexicographically and chooses the lexicographically smaller node.



	P1	P2	P3
Lex1	17.2	82.8	0.0
Lex2	17.8	82.2	0.0
Avg	47.2	51.3	1.5
Min	27.9	68.6	3.5
Max	43.1	56.9	0.0

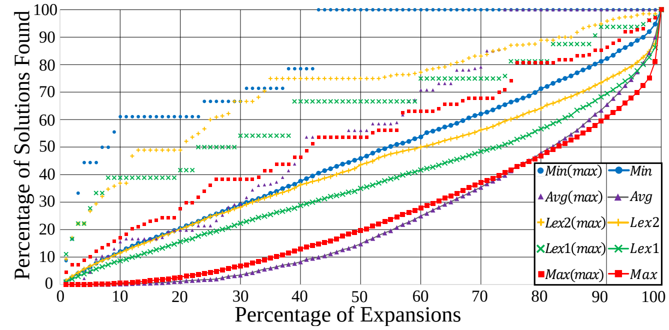


Figure 1: (Left) Areas of nodes in MOS; (Center) %expansions in each phase; (Right) %solutions for %expansions.

Expansion Phases of the Search

Regardless of which Ordering Function is used, best-first search algorithms have to expand all the MENs, some of the MBENs, and none of the NENs. However, they can find solutions in a different order. That is, each Ordering Function finds the solutions in the same order it prioritizes nodes. Hence, we divide the search into three expansion phases. P1: Nodes that are expanded before the first solution was found. P2: Nodes that are expanded after finding the first solution but before finding the last solution (i.e., before finding the entire POF). P3: Nodes that are expanded after finding the last solution. P3 does not exist in *Lex* (for any *Lex* Ordering Function) and *Max*, because any node that is prioritized after the last solution cannot be a MEN. *Min* and *Avg* might find all the solutions before finishing the search.

Empirical Evaluation

We evaluated the average percentage of expansions of each Ordering Function in the different expansion phases defined above on 200 random instances of the BAY road-map (DIMACS 2006). For heuristics, we used the common *Point Heuristic* which takes the shortest path of each objective individually. The table in Fig. 1(Center) presents the results. In P1, as expected, *Lex1* and *Lex2* have a relatively small percentage. This is because they find the (lexicographically) first solution with a perfect heuristic in one dimension. *Min* finds the first solution after simulating both *Lex* functions while only expanding overlapping nodes once. So, it expands slightly less than the sum of the two *Lex* functions. Finally, *Avg* and *Max* expanded almost 50% of the nodes before finding the first solution. In P2, nodes are expanded and new solutions are found. In P3, the nodes are expanded to prove that there are no more solutions in the POF. *Min* expanded 3.5% of the nodes in P3, which means that *Min* found the entire POF the fastest, as all Ordering Functions expanded the exact same number of nodes in each instance.

Fig. 1(Right), presents the percentage of solutions found from the POF (y-axis) as a function of the percentage of expansions that have passed (x-axis) until the search halts for each Ordering Function (*Min*, *Lex2*, *Avg*, *Lex1*, *Max*). In the figure, closer to the top-left is better because in the top-left area more solutions are found faster. Also, for each function, we measured the maximal value (*Min*(max), *Avg*(max), *Lex2*(max), *Lex1*(max), *Max*(max)). Namely, for each percentage of expansions, we present the highest value each function achieved. On average, *Max* and *Avg* performed worst, then *Lex1* and *Lex2*, and the best performance was achieved by *Min*. By observing the maximum values achieved by each function, we can see a correlation with the average results. As mentioned, the last node explored by *Lex* and *Max* is a solution. Therefore, these functions did not reach 100% of the solutions before expanding all 100% of the nodes. In contrast, we can see that there was a case in which *Avg* was able to find all POF after 73% of the expansions and *Min* was able to find all solutions after only 43% of the expansions.

To summarize, while *Lex* reaches the first solution the fastest, other functions (*Avg* and *Min*) are able to find all POF faster, with fewer node expansions.

Acknowledgments

This research was supported by the US-Israel Binational Science Foundation (BSF) grant no. 2021643 and by NSF grant no. 2121028.

References

Clímaco, J.; and Pascoal, M. 2012. Multicriteria Path and Tree Problems: Discussion on Exact Algorithms and Applications. *itor*, 19(1-2): 63–98.

Dechter, R.; and Pearl, J. 1985. Generalized Best-First Search Strategies and the Optimality of A^*. *J. ACM*, 32(3): 505–536.

DIMACS. 2006. The 9th DIMACS Implementation Challenge: Shortest Path.

Eckerle, J.; Chen, J.; Sturtevant, N. R.; Zilles, S.; and Holte, R. C. 2017. Sufficient Conditions for Node Expansion in Bidirectional Heuristic Search. In *ICAPS*, 79–87.

Hernandez, C. U.; Yeoh, W.; Baier, J. A.; Zhang, H.; Suazo, L.; and Koenig, S. 2020. A Simple and Fast Bi-Objective Search Algorithm. In *ICAPS*, 143–151.

Mandow, L.; and De La Cruz, J. L. P. 2005. A New Approach to Multi-Objective A^* Search. In *IJCAI*, 218–223.

Skriver, A. J. 2000. A Classification of Bicriterion Shortest Path (BSP) Algorithms. *Asia Pacific J. of Oper. Res.*, 17(2): 199–212.

Stewart, B. S.; and White III, C. C. 1991. Multiobjective A^*. *Journal of the ACM*, 38(4): 775–814.