Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Survey Track

Heuristic-Search Approaches for the Multi-Objective Shortest-Path Problem: Progress and Research Opportunities

Oren Salzman[1] , **Ariel Felner**[2] , **Carlos Hernandez**[3] , **Han Zhang**[4] , **Shao-Hung Chan**[4] , **Sven Koenig**[4]
[1]Technion—Israel Institute of Technology
[2]Ben-Gurion University of the Negev
[3]Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Santiago, Chile
[4]University of Southern California

Abstract

In the multi-objective shortest-path problem we are interested in computing a path, or a set of paths that simultaneously balance multiple cost functions. This problem is important for a diverse range of applications such as transporting hazardous materials considering travel distance and risk. This family of problems is not new with results dating back to the 1970's. Nevertheless, the significant progress made in the field of heuristic search resulted in a new and growing interest in the sub-field of multi-objective search. Consequently, in this paper we review the fundamental problems and techniques common to most algorithms and provide a general overview of the field. We then continue to describe recent work with an emphasis on new challenges that emerged and the resulting research opportunities.

1 Introduction

Many optimization problems involve multiple objectives that should be optimized. For example, when planning a driving route, one may wish to minimize both driving distance as well as tolls. Alternatively, when planning power line routes one may wish to consider both economic and ecological impacts Such problems have been used in numerous applications where decisions need to be taken when it is not possible to simultaneously optimize the different objectives [Miettinen, 2012; Roijers and Whiteson, 2017; Hwang and Masud, 2012; Emmerich and Deutz, 2018]. Multiobjective optimization typically consists of an optimization phase in which a set of possible solutions is computed and a decision-making phase in which a decision maker chooses a solution computed in the optimization phase [Roijers and Whiteson, 2017].

In this paper we restrict ourselves to the optimization phase that occurs in a very specific instance of multiobjective-optimization problems termed the *multi-objective shortest-path problem*, a generalization of the well-studied shortest-path problem. This family of problems is not new with results dating back to the 1970's [Vincke, 1976] and several surveys on the topic appeared [Clímaco and Pascoal, 2012; Current and Marsh, 1993; Skriver, 2000; Tarapata, 2007; Ulungu and Teghem, 1991].

Nevertheless, the field of heuristic search has made significant progress which resulted in a new and growing interest in the sub-field of multi-objective search. To this end, we present an overview of the field with an emphasis on results obtained in recent years using heuristic-search techniques. Importantly, other approaches which are out of the scope of this paper have been used. These include evolutionary algorithms (see, e.g., [Pangilinan and Janssens, 2007; Li *et al.*, 2015]), integer programing (see, e.g., [Halffmann *et al.*, 2022]), as well as reinforcement-learning algorithms (see, e.g., [Tozer *et al.*, 2017; Hayes and others, 2022]).

Perhaps the most closely-related books and reviews are on the topic of multi-objective optimization in sequential decision problems, commonly modeled as Markov decision processes, which can be seen as a generalization of our problem [Roijers *et al.*, 2013; Roijers and Whiteson, 2017]. However, restricting the problem to multi-objective shortest-path computation opens the door for highly-efficient algorithms as well as new algorithmic challenges. Advances in our setting can then be used to advance the field of multi-objective optimization in sequential decision problems.

Outline We begin in Sec. 2 by introducing notation and our problem definition and continue in Sec. 3 to describe general algorithmic building blocks common to many state-of-the-art algorithms. Subsequently, we start with a general overview of relevant work in Sec. 4 followed by a deep-dive into recent work in Sec. 5 with an emphasis on new challenges. We then move on (Sec. 6) to describe commonly-used benchmarks and conclude (Sec. 7) with a discussion and summary.

2 Notation and Setting

In this section, we introduce relevant notation (Sec. 2.1) and the multi-objective shortest-path problem (Sec. 2.2).

2.1 Notation

As we will see, paths in our setting will have a cost value for each objective. Thus, a common practice is to associate cost vectors with paths. Here we introduce the relevant notation.

Boldface font indicates vectors, lower-case and upper-case symbols indicate elements and sets, respectively. The notation p_i will be used to denote the i'th component of $\mathbf{p}$. The

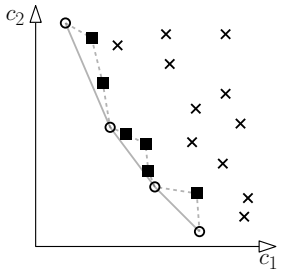

Figure 1: Types of solutions. Dominated solutions that are not part of Π^* are denoted by $\times$, Solutions that belong to Π^* and are and aren't extreme (see Sec. 4.2) are denoted by $\bigcirc$ and $\blacksquare$, respectively.

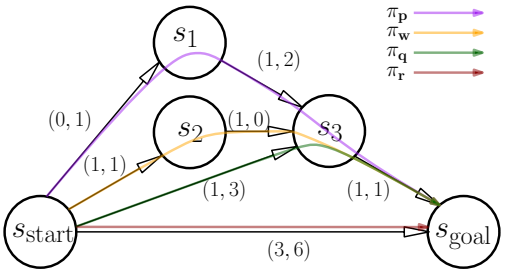

Figure 2: An illustrative bi-objective search graph with four solutions connecting s_{start} to s_{goal}. Figure best viewed in color.

addition of two d-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$ and the multiplication of a real-valued scalar k and a d-dimensional vector $\mathbf{p}$ are defined in the natural way, namely as $\mathbf{p} + \mathbf{q} = (p_1+q_1, \ldots, p_d+q_d)$ and $k\mathbf{p} = (kp_1, \ldots, kp_d)$, respectively.

Let $\mathbf{p}$ and $\mathbf{q}$ be d-dimensional vectors. We say that $\mathbf{p}$ *weakly dominates* $\mathbf{q}$ and denote this as $\mathbf{p} \preceq \mathbf{q}$ if $\forall i, p_i \leq q_i$. We say that $\mathbf{p}$ *dominates* $\mathbf{q}$ and denote this as $\mathbf{p} \prec \mathbf{q}$ if $\mathbf{p}$ weakly dominates $\mathbf{q}$ and $\exists j, p_j < q_j$. Similarly, when $\mathbf{p}$ doesn't dominate (resp. doesn't weakly dominate) $\mathbf{q}$, we will write $\mathbf{p} \not\prec \mathbf{q}$ (resp. $\mathbf{p} \not\preceq \mathbf{q}$). For $\mathbf{p} \neq \mathbf{q}$, if $\mathbf{p} \not\preceq \mathbf{q}$ and $\mathbf{q} \not\preceq \mathbf{p}$ we say that $\mathbf{p}$ and $\mathbf{q}$ are *mutually undominated*. Finally, we say that $\mathbf{p}$ is *lexicographically smaller* than $\mathbf{q}$ and denote this as $\mathbf{p} \prec_{\text{lex}} \mathbf{q}$ if $\mathbf{p}_k < \mathbf{q}_k$ for the first index k s.t. $\mathbf{p}_k \neq \mathbf{q}_k$.

Let $\mathbf{X}$ be a set of d-dimensional vectors. We say that $\mathbf{X}$ is a *mutually undominated set* if all pairs of vectors in $\mathbf{X}$ are mutually undominated (namely, no vector in $\mathbf{X}$ weakly dominates any other vector in $\mathbf{X}$). Now, given a (not-necessarily mutually undominated) set $\mathbf{X}$, $\mathcal{MU}(\mathbf{X})$ is a cost-unique subset of $\mathbf{X}$ that is a mutually undominated set.[1] Note that $\mathcal{MU}(\mathbf{X}) \subseteq \mathbf{X}$ and that $\mathcal{MU}(\mathbf{X})$ may not be unique.

Finally, let $\mathbf{p}$ and $\mathbf{q}$ be two d-dimensional vectors and let ε be another d-dimensional vector such that $\forall i\ \varepsilon_i \geq 0$. We say that $\mathbf{p}$ *approximately dominates* $\mathbf{q}$ with an *approximation factor* ε and denote this as $\mathbf{p} \preceq_\varepsilon \mathbf{q}$ if $\forall i, p_i \leq (1+\varepsilon_i) \cdot q_i$.

Example 1. *Consider the following two-dimensional vectors* $\mathbf{p} = (2,4)$, $\mathbf{q} = (2,4)$, $\mathbf{r} = (3,6)$ *and* $\mathbf{w} = (3,2)$ *Here,* $\mathbf{p} \preceq \mathbf{q}$ *(i.e.,* $\mathbf{p}$ *weakly dominates* $\mathbf{q}$*),* $\mathbf{p} \prec \mathbf{r}$ *(i.e.,* $\mathbf{p}$ *dominates* $\mathbf{r}$*) and both* $\mathbf{p} \not\prec \mathbf{w}$ *and* $\mathbf{w} \not\prec \mathbf{p}$ *(i.e.,* $\mathbf{p}$ *and* $\mathbf{w}$ *are mutually undominated). Similarly, for* $\varepsilon = (1,2)$ *we have that* $\mathbf{r} \preceq_\varepsilon \mathbf{p}$ *(i.e.,* $\mathbf{r}$ *approximately dominates* $\mathbf{p}$*). Finally, we have that* $\mathbf{p} \prec_{\text{lex}} \mathbf{w} \prec_{\text{lex}} \mathbf{r}$ *and if we set* $\mathbf{X} = \{\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{w}\}$*, then* $\mathcal{MU}(\mathbf{X})$ *can be either* $\{\mathbf{p}, \mathbf{w}\}$ *or* $\{\mathbf{q}, \mathbf{w}\}$.

2.2 Setting

A multi-objective *search graph* is a tuple $(S, E, \mathbf{c})$, where S is the finite set of *states*, $E \subseteq S \times S$ is the finite set of edges, and $\mathbf{c} : E \to \mathbb{R}^d_{\geq 0}$ is a *cost function* that associates a d-dimensional vector of non-negative real costs with each edge. A *path* π from s_1 to s_n is a sequence of states $s_1, s_2, \ldots, s_n$ such that $(s_i, s_{i+1}) \in E$ for all $i \in \{1, \ldots, n-1\}$.

We define the *cost* of a path $\pi = s_1, \ldots, s_n$ as $\mathbf{c}(\pi) = \sum_{i=1}^{n-1} \mathbf{c}(s_i, s_{i+1})$. Given paths π and π', we extend all the above definitions to paths. E.g., we say that π *dominates* (resp. *weakly dominates*) π' and denote this as $\pi \prec \pi'$ (resp. $\pi \preceq \pi'$) if $\mathbf{c}(\pi) \prec \mathbf{c}(\pi')$ (resp. $\mathbf{c}(\pi) \preceq \mathbf{c}(\pi')$).

Let $(S, E, \mathbf{c})$ be a multi-objective search graph and $s_{\text{start}} \in S$ a start state. The *complete Pareto-optimal frontier* of s, denoted as $\tilde{\Pi}_s$, is the set of all paths from s_{start} to s that are not weakly dominated by any other path from s_{start} to s. As $\tilde{\Pi}_s$ may contain multiple paths with the same cost, we are often interested in a subset of $\tilde{\Pi}_s$ which contains only mutually-undominated paths. We call such a set the cost unique pareto optimal frontier or just *Pareto-optimal frontier* for short.

Typically, we are also given a goal state $s_{\text{goal}} \in S$ and we call a path from s_{start} to s_{goal} a *solution* and its Pareto-optimal frontier, which we denote Π^*, as the *Pareto-optimal solution set*. See Fig. 1. Intuitively, Π^* contains all "interesting" paths—those that are not dominated by any other solution and may be candidate solutions that the decision maker will choose from. However, computing Π^* is NP-hard [Serafini, 1987] as its cardinality may be exponential in $|S|$ [Hansen, 1980; Ehrgott, 2005; Breugem *et al.*, 2017]. Even determining whether a path belongs to Π^* is NP-hard [Papadimitriou and Yannakakis, 2000].

In real-world settings we are often not interested in the entire Pareto front—it's too large to present to decision makers. Thus, we are often interested in computing a bounded approximation of Π^*. To this end, given an approximation factor ε, an *ε-approximate Pareto-optimal solution set* Π^*_ε is a set of solutions such that every path in Π^* is ε-dominated by a path in Π^*_ε. Note that (i) the ε-approximate Pareto-optimal solution set is not necessarily unique and that (ii) Some variants of this definition require that $\Pi^*_\varepsilon \subseteq \Pi^*$ while others don't. Namely, they allow Π^*_ε to contain solutions that do not belong to Π^* (but still approximately dominating solutions in Π^*).

Example 2. *Consider the graph presented in Fig. 2 that contains the following solutions* $\pi_p = \langle s_{\text{start}}, s_1, s_3, s_{\text{goal}} \rangle$ *with* $\boldsymbol{c}(\pi_p) = (2,4)$, $\pi_q = \langle s_{\text{start}}, s_3, s_{\text{goal}} \rangle$ *with* $\boldsymbol{c}(\pi_q) = (2,4)$, $\pi_r = \langle s_{\text{start}}, s_{\text{goal}} \rangle$ *with* $\boldsymbol{c}(\pi_r) = (3,6)$ *and* $\pi_w = \langle s_{\text{start}}, s_2, s_3, s_{\text{goal}} \rangle$ *with* $\boldsymbol{c}(\pi_w) = (3,2)$*. Note that solution costs correspond to the vectors used in Example 1. Here, the complete Pareto-optimal frontier* $\tilde{\Pi}_{s_3}$ *to* s_3 *contains all three paths to* s_3 *while a Pareto-optimal frontier to* s_3 *contains the paths that traverse* s_1 *and* s_2*. Finally, a Pareto-optimal solution set* Π^* *is* $\{\pi_p, \pi_w\}$*, and for an approximation factor of* $\varepsilon = (1,2)$ *an* ε*-approximate Pareto-optimal solution set* Π^*_ε *can be* $\{\pi_p\}$*, or* $\{\pi_r\}$*,*

[1] A set $\mathbf{X}$ is said to be cost unique if $\forall \mathbf{p}, \mathbf{q} \in \mathbf{X}\ \exists k\ s.t.\ \mathbf{p}_k \neq \mathbf{q}_k$.

3 Algorithmic Problems and Building Blocks

In this section, we describe the fundamental problems that multi-objective search algorithms typically need to solve (Sec. 3.1) as well as key algorithmic building blocks used by many state-of-the-art algorithms (Sec. 3.2 and 3.3).

3.1 Dominance Checks

Recall that in the single-objective shortest-path problem we are only required to maintain one path to each state (as one path weakly dominates all other paths found so far). In contrast, in the multi-objective case we often need to store a mutually undominated set of paths. When an algorithm considers a new path, we need to check if it is dominated by any of the paths found so far. If this is not the case, we need to update the mutually undominated set of paths maintained, possibly deleting an existing path if it is dominated by the new path. These operations, which are key in many algorithms, are formalized in the following two problems, which pinpoint the additional complexity inherent in multi-objective search.

Problem 1 (Dominance Check (DC))**.** *Given a mutually undominated set $\mathbf{X}$ of d-dimensional vectors (with $d \geq 2$) and a new d-dimensional vector $\mathbf{p}$, the DC problem calls to verify whether there exists a vector $\mathbf{q} \in \mathbf{X}$ such that $\mathbf{q} \preceq \mathbf{p}$.*

Problem 2 (Mutually Undominated Set Update (MUSU))**.** *Given a mutually undominated set $\mathbf{X}$ of d-dimensional vectors (with $d \geq 2$) and a new d-dimensional vector $\mathbf{p}$ that is not dominated by any vector in $\mathbf{X}$, the MUSU problem computes $\mathcal{MU}(\mathbf{X} \bigcup \{\mathbf{p}\})$.*

Naive solutions to the DC and MUSU problem may require $O(|\mathbf{X}|)$ vector comparisons in the worst case. However, data-structures such as ND-Trees [Jaszkiewicz and Lust, 2018] have been specifically tailored to improve the efficiency of these problems (see, e.g. [Fieldsend, 2020; Altwaijry and Menai, 2012]). Interestingly, these data structures are typically employed by evolutionary algorithms and not heuristic-search algorithms.

Example 3. *Consider the following two-dimensional vectors used in Example. 1 $\mathbf{p} = (2, 4)$, $\mathbf{q} = (2, 4)$, $\mathbf{r} = (3, 6)$ and $\mathbf{w} = (3, 2)$ and set $\mathbf{X} = \{\mathbf{p}, \mathbf{r}\}$. Solving the DC problem for both $\mathbf{q}$ and $\mathbf{w}$ on $\mathbf{X}$ yields that $\mathbf{p} \in \mathbf{X}$ dominates $\mathbf{w}$ and that there is no element in $\mathbf{X}$ that dominates $\mathbf{w}$. Now, if we solve the MUSU problem for $\mathbf{w}$ on $\mathbf{X}$ we can conclude that $\mathcal{MU}(\mathbf{X} \bigcup \{\mathbf{w}\}) = \{\mathbf{p}, \mathbf{w}\}$.*

3.2 Dimensionality Reduction

The following approach, first presented by Pulido *et al.* [2015] and termed *dimensionality reduction*, is a general technique that helps to solve the DC problem (Prob. 1). However, to describe it, we need to introduce some additional notation. For a d-dimensional vector $\mathbf{p} = (p_1, \ldots, p_d)$, its truncated vector $\mathrm{TR}(\mathbf{p})$ is a $(d-1)$-dimensional vector without $\mathbf{p}$'s first component, i.e., $\mathrm{TR}(\mathbf{p}) := (p_2, \ldots, p_d)$. Similarly, given a set of vectors $\mathbf{X}$, its associated set of truncated vectors is $\mathrm{TR}(\mathbf{X}) := \{\mathrm{TR}(\mathbf{p}) | \mathbf{p} \in \mathbf{X}\}$

Let $\mathbf{X}$ be a set of vectors and $\mathbf{p}$ be a vector lexicographically larger than all elements in $\mathbf{X}$. Namely, $\forall \mathbf{q} \in \mathbf{X}$, $\mathbf{q} \prec_{\mathrm{lex}} \mathbf{p}$. Then, $\mathbf{p}$ is weakly dominated by a vector in $\mathbf{X}$ iff $\exists \mathbf{q} \in \mathcal{MU}(\mathrm{TR}(\mathbf{X}))$ for which $\mathrm{TR}(\mathbf{q}) \preceq \mathrm{TR}(\mathbf{p})$.

We can complete the DC test in $O(|\mathcal{MU}(\mathrm{TR}(\mathbf{X}))|)$ time instead of $O(|\mathbf{X}|)$ time. As the cardinality of $\mathcal{MU}(\mathrm{TR}(\mathbf{X}))$ is often much smaller than that of $\mathbf{X}$, the procedure just described often requires far less comparison operations than naive solutions to Prob. 1. Moreover, for the specific setting where we have two objectives, $|\mathcal{MU}(\mathrm{TR}(\mathbf{X}))| = 1$ (as it is the minimum of the second component of all elements in $\mathbf{X}$), regardless of $|\mathbf{X}|$ and the test takes $O(1)$ time.

Example 4 (adapted from [Pulido *et al.*, 2015])**.** *Consider the set $\mathbf{X} = \{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$ with $\mathbf{p} = (6, 2, 4)$, $\mathbf{q} = (4, 4, 5)$ and $\mathbf{r} = (2, 3, 6)$ and note that $\mathbf{X} = \mathcal{MU}(\mathbf{X})$ and $\mathcal{MU}(\mathrm{TR}(\mathbf{X})) = \{\mathrm{TR}(\mathbf{p})\}$. To test if $\mathbf{w} = (7, 2, 4)$ is dominated by a vector in $\mathbf{X}$ (not using dimensionality reduction) may require up to three vector comparisons but because $|\mathcal{MU}(\mathrm{TR}(\mathbf{X}))| = 1$ and $\mathrm{TR}(\mathbf{p}) \preceq \mathrm{TR}(\mathbf{w})$, we only need one vector comparison to conclude that $\mathbf{w} \notin \mathcal{MU}(\mathbf{X} \bigcup \{\mathbf{w}\})$.*

3.3 Heuristics

Similar to single-objective search algorithms, multi-objective search algorithms often use a heuristic function to guide the search. A *single-valued heuristic* $\mathbf{h} : S \rightarrow \mathbb{R}^d_{\geq 0}$ estimates the cost to s_{goal} from every state s.[2] We say that $\mathbf{h}$ is *admissible* iff $\forall s \in S$, $\mathbf{h}(s) \preceq \mathbf{c}(\pi)$ for any path π from s to s_{goal}. Similarly, $\mathbf{h}$ is *consistent* iff $\mathbf{h}(s_{\mathrm{goal}}) = (0, \ldots, 0)$ and $\mathbf{h}(s) \leq \mathbf{c}(s, s') + \mathbf{h}(s')$ for all $(s, s') \in E$.

For any path π_s from s_{start} to some state s, if $\mathbf{h}$ is admissible and if there exists a path $\pi_{\mathrm{goal}} \in \Pi^*$ such that $\mathbf{c}(\pi_{\mathrm{goal}}) \preceq \mathbf{c}(\pi_s) + \mathbf{h}(s)$, then any path extending π_s to s_{goal} will be dominated by π_{goal}. If such a setting is identified by a search algorithm, it can discard π_s. We call this *goal pruning* (other pruning techniques exist that do not use heuristics).

Unless stated otherwise (see Sec. 5.1), existing algorithms use the so-called "ideal point heuristic" $\mathbf{h}_{\mathrm{ideal}}$ which combines a set of d single-objective heuristics $h_1, \ldots, h_d$. Here, $h_i : S \rightarrow \mathbb{R}_{\geq 0}$ corresponds to the shortest path from each state according to the i'th objective and $\forall s \in S$ $\mathbf{h}_{\mathrm{ideal}}(s) := (h_1(s), \ldots, h_d(s))$. The ideal point heuristic, which is admissible, is easily computed by running d (single-objective) instances of Dijkstra's algorithm starting from s_{goal} (i.e., one instance for each objective).

Example 5. *The ideal point heuristic $\mathbf{h}_{\mathrm{ideal}}$ for state s_{start} in the graph depicted in Fig. 2 is $(2, 2)$.*

4 Existing Algorithms—Overview

In this section we start by reviewing algorithms to compute Π^* (Sec. 4.1), and subsets of Π^* (Sec. 4.2). We conclude (Sec. 4.3) by describing algorithms for variants of the multi-objective shortest-path problem.

4.1 Efficient Computation of Π^*

To efficiently compute Π^*, both generalizations of the label-correcting paradigm to the multi-objective setting were suggested [Guerriero and Musmanno, 2001] as well as adapta-

[2] The reason we call this a single-valued heuristic and not simply a heuristic will be evident in Sec. 5.1.

tions of the celebrated A* algorithm [Hart *et al.*, 1968]. Here, we focus on the latter approach.

Stewart et al. (1991) introduced Multi-Objective A* (MOA*) which is a multi-objective generalization of A*. Similar to A*, MOA* maintain a priority queue OPEN, which contains the generated but not yet expanded nodes. MOA* also stores a set of solution nodes (A* does not do this because it halts once the first solution is found). Each node n contains a state $s(n)$, a $\mathbf{g}$-value $\mathbf{g}(n)$ which is the cost to reach $s(n)$ from s_{start} along a path $\pi(n)$, an $\mathbf{h}$-value $\mathbf{h}(n)$, and an $\mathbf{f}$-value $\mathbf{f}(n) = \mathbf{g}(n) + \mathbf{h}(n)$. At each iteration, MOA* extracts a node n that is not dominated by any other node in OPEN (similar to using the minimal $\mathbf{f}$-value in A*). It then solves the DC problem to determine if $\mathbf{g}(n)$ is weakly dominated by the $\mathbf{g}$-value of some other node n' with $s(n) = s(n')$ or if $\mathbf{f}(n)$ is weakly dominated by the $\mathbf{f}$-value of a solution node. If so, the node is pruned. If n is extracted and not pruned, MOA* expands it by either generating its child nodes if $s(n) \neq s_{\text{goal}}$ or adding it to the solution set by solving the MUSU problem. When OPEN becomes empty, MOA* terminates and returns the solution set.

MOA* served as the foundation to multiple extensions (see, e.g., [Mandow and De La Cruz, 2005; Mandow and De La Cruz, 2010]) which differ in which information is contained in the nodes, how nodes are ordered in OPEN, and how dominance checks are implemented and at what stages they are performed (upon generation or upon expansion). Unfortunately, as we discuss in Sec. 6, different benchmarks are used and it is not always clear if one algorithm always outperforms other state-of-the-art methods.

A key insight that was used to dramatically improve the performance of these algorithms was to order nodes in OPEN in increasing lexicographic order and apply the notion of dimensionality reduction introduced by Pulido *et al.* [2015] and described in Sec. 3. This was used to suggest a multi-objective search algorithm named NAMOA-dr[3]. Hernández *et al.* [2023a] adapted and simplified NAMOA-dr for the bi-objective setting to suggest the Bi-Objective A* algorithm (BOA*) which allows to perform dominance checks in $O(1)$ time when a consistent heuristic is used.

Ahmadi *et al.* [2021] extended BOA* by simultaneously running two BOA*-like searches from s_{start}—one where the lexicographical order is done according to (c_1, c_2) and the second where the lexicographical order is done according to (c_2, c_1). Ren *et al.* [2022c] presented Enhanced Multi-Objective A* (EMOA*) which efficiently maintains Pareto-optimal frontiers for multiple (i.e., more than two) objectives by incrementally constructing balanced binary search trees within the MOA* search framework

Sedeño-Noda and Colebrook [2019] introduced an adaption of Dijkstra's algorithm to the bi-objective setting named Bi-objective Dijkstra or BDA. Key to the efficiency of BDA is that one candidate node is stored for each state at any given point of time. This dramatically simplifies the complexity of the DC and MUSU problems solved within BDA. Consequently, the theoretical running time of BDA is $|\Pi^*|$ multiplied by Dijkstra's running time.

[3]Here, 'dr' stands for dimensionality reduction.

BDA was recently extended to the multi-objective setting. Specifically, de las Casas *et al.* [2021b] used both a heuristic and an upper-bound on the cost to reach the target in order to improve running times. While de las Casas *et al.* [2021c] suggested a memory-efficient multi-objective shortest-path algorithm generalizing BDA. Here, the size of the priority queue used in the algorithm is bounded by $|S|$. This yields an output-sensitive running time bound for the new algorithm that is roughly the running time of Dijkstra's algorithm.

4.2 Efficient Computation of Subsets of Π^*

Following the aforementioned hardness results, and as we are often only interested in one or a few solutions that lie on, or close to Π^*, an active line of work has been concerned with computing only a subset of Π^* or computing Π^*_ε.

Bounded Approximations of Π^*

Early approaches towards approximating Π^* focused on Fully Polynomial Time Approximation Schemes[4] (FPTAS) [Vazirani, 2001]. Warburton [1987] used scaling and rounding techniques while Perny and Spanjaard [2008] presented another FPTAS assuming that a finite upper bound L on the numbers of arcs of all solutions in Π^* is known. This assumption was later relaxed [Tsaggouris and Zaroliagis, 2009; Breugem *et al.*, 2017] by partitioning the space of solution into a grid of cells whose dimensions are a function of the approximation factor and, roughly speaking, take only one solution in each grid cell. Recently de las Casas *et al.* [2021a] suggest an FPTAS for the dynamic multi-objective shortest-path problem in which edges undergo cost changes. Unfortunately, the complexity of FPTASs is typically bounded by a polynomial of high degree, and hence they are often slower in practice than approaches to compute Π^* even when applied to relatively-small instances. Consequentally, running these approaches on moderately-sized graphs (i.e., with roughly $10,000$ states) is often impractical [Breugem *et al.*, 2017].

An alternative approach to compute Π^*_ε (though not necessarily in polynomial time) was suggested by Perny and Spanjaard [2008]. They suggest a simple variation of MOA*, termed MOA*$_\varepsilon$ that reduces computation times by punning intermediate paths that are approximately dominated by already-computed solutions. Subsequently, Goldin and Salzman [2021] suggested PPA*, an extension of BOA* that outperforms MOA*$_\varepsilon$ by introducing new pruning techniques. This was later generalized by Zhang *et al.* [2022b] to allow approximation of Π^* for any number of objectives. Noteworthy is that while they all return solutions that approximate Π^*, MOA*$_\varepsilon$ returns solutions that belong to Π^* while PPA* may return solutions not on Π^*.

General Subsets of Π^*

A different approach to compute a subset of Π^* is to compute the set of *extreme solutions*, also known as the set of *supported solutions* [Sedeño-Noda and Raith, 2015]. Formally, the set of extreme solutions Π^*_{extreme} is defined as the set of

[4]An FPTAS is an approximation scheme whose time complexity is polynomial in the input size and also polynomial in $1/\varepsilon$ where ε is the approximation factor.

solutions whose cost lie on the convex hull of Π^*. For a visualization, see Fig. 1.

Interestingly, any extreme solution corresponds to finding a solution that minimizes a linear combination of the different objectives [Roijers *et al.*, 2013]. Thus, computing the set of extreme solutions is useful when we know that the desired solution is a linear combination of the different objectives but we don't know a-priori which weights will be used and want to have all options available. Approaches to compute Π^*_{extreme} for the bi-objective setting include a Dijkstra-like method [Sedeño-Noda and Raith, 2015] as well as a branch-and-bound-based approach that only executes a series of (single-objective) shortest-path searches [Zhang *et al.*, 2016]. Noteworthy is that $|\Pi^*_{\text{extreme}}|$ may be orders of magnitude smaller than $|\Pi^*|$. E.g., when comparing the two on road maps from the 9'th DIMACS Implementation Challenge: Shortest Path (see Sec. 6), $|\Pi^*|$ is larger (on average) than $|\Pi^*_{\text{extreme}}|$ by $17\times$ on the small NY instance and by $168\times$ on the large LKS instance.

Taking a different approach to approximate Π^*, Legriel et al. (2010) suggest a method based on satisfiability/constraint solvers. Alternatively, Rivera *et al.* [2022] consider the bi-objective setting and suggest to transform the original bi-objective search problem $\mathcal{P}$ into another bi-objective problem $\mathcal{P}'$ such that the number of solutions of $\mathcal{P}'$ is smaller (potentially orders of magnitude smaller, depending on values of the algorithm's hyperparameters) than the number of solutions of $\mathcal{P}$. Importantly, each solution to $\mathcal{P}'$ is a solution to $\mathcal{P}$ (though not vice-versa).

4.3 Variants of Multi-Objective Shortest-Path

Skyler *et al.* [2022] consider the setting where we want a solution which belongs to Π^* whose costs are below given upper bounds on each objective. This is a variant of the constrained shortest-path problem (CSP) [Storandt, 2012]. Zhang *et al.* [2022a] suggest an anytime algorithm for the bi-objective setting such that whenever the algorithm is terminated, it outputs an approximate Pareto-optimal solution set.

Ren *et al.* [2022b] consider the problem of finding collision-free Pareto-optimal solutions for agents moving amid obstacles that follow known trajectories while simultaneously optimizing multiple objectives. Ren *et al.* [2022a] propose an approach based on D* Lite [Koenig and Likhachev, 2002] that allows interleaving planning and execution in the multi-objective case while Ren *et al.* [2021] propose to extend the multi-agent path finding problem (MAPF) [Stern *et al.*, 2019; Salzman and Stern, 2020] to the multi-objective case.

5 Recent Advances and Open Challenges

In this section, we perform a deeper-dive into several topics in multi-objective search with an emphasis on new emerging challenges in the field.

5.1 Single-Valued vs. Multi-Valued Heuristics

Following Mandow and De La Cruz [2010] and in contrast to the single-objective setting, in the general case of multi-objective search, the heuristic value of a state s is not a single number, but a *set* of *cost vectors*. Formally, a *multi-valued* heuristic is defined as a mapping $H : S \rightarrow 2^{\mathbb{R}^d_{\geq 0}}$. The notions of admissibility and consistence introduced for single-valued heuristics are naturally extended to multi-valued heuristics as well as the notion of goal pruning which we now detail.

A multi-valued heuristic H is admissible iff for every state s and every path π_s from s to s_{goal}, H contains a cost vector that weakly dominates $\mathbf{c}(\pi_s)$.

Example 6. *Consider the graph presented in Fig. 2 and the single-valued heuristics h_1 and h_2 s.t. $h_1(s_{\text{start}}) = (2, 4)$, $h_2(s_{\text{start}}) = (3, 2)$, $h_1(s_1) = h_2(s_1) = (2, 3)$, $h_1(s_2) = h_2(s_2) = (2, 2)$ and $h_1(s_3) = h_2(s_3) = (1, 1)$. If we set $H = \{h_1, h_2\}$, then it is easy to verify that H is admissible*

Goal Pruning Using Multi-Valued Heuristics

Let n be a node in a search algorithm corresponding to some path $\pi(n)$ leading to a state $s(n)$. Furthermore, assume that we are using an admissible multi-valued heuristic H and that Π_{goal} is the current mutually-undominated set of solutions found by the algorithm. The node n can be pruned if for every single-values heuristic $\mathbf{h} \in H$ there exists a solution $\pi_{\text{goal}} \in \Pi_{\text{goal}}$ that weakly dominates n's $\mathbf{f}$-value when computed according to $\mathbf{h}$. Namely, n can be pruned if

$$\forall \mathbf{h} \in H, \exists \pi_{\text{goal}} \in \Pi_{\text{goal}} \text{ s.t. } \mathbf{c}(\pi_{\text{goal}}) \preceq \mathbf{c}(\pi(n)) + \mathbf{h}(s(n)).$$

Testing for goal pruning can be easily implemented in $O(d \cdot |H| \cdot |\Pi_{\text{goal}}|)$ time. However, for the case where a single-valued heuristic is used (i.e., when $|H| = 1$) recent results suggest how to do this more efficiently: As mentioned in Sec. 4.1, Hernández *et al.* [2023a] describe a method to test for goal pruning in constant time when considering only two objectives and Ren *et al.* [2022c] show that testing for goal pruning can be done in $O(\log(|\Pi_{\text{goal}}|))$ time when considering three objectives and in $O((d-1) \cdot |\Pi_{\text{goal}}|)$ for an arbitrary number of objectives d. However, it is not clear how to extend these results when using multi-valued heuristics.

Heuristics Used in Empirical Evaluations

Typical empirical evaluation (see e.g., [Stewart and White III, 1991; Hernández *et al.*, 2023a] and Sec. 6 below) only make use of the "ideal point heuristic" $\mathbf{h}_{\text{ideal}}$. The only work that empirically considers the implication of using a multi-valued heuristic is the recent paper by Geißer *et al.* [2022] in which they study the question of how to construct informative multi-valued admissible heuristics. Specifically, they generalized several classes of well-known admissible planning heuristics to the multi-objective case . Namely abstraction-based (see, e.g., [Sievers and Helmert, 2021]), critical path-based (see, e.g., [Haslum and Geffner, 2000]), and LP/IP-based heuristics (see, e.g., [Pommerening *et al.*, 2014].Interestingly, on the benchmarks tested, these heuristics are often more informed than $\mathbf{h}_{\text{ideal}}$.[5] Having said that, on other benchmarks test, either this is not true or the overhead of computing these heuristics is large and the total runtime is larger than when using $\mathbf{h}_{\text{ideal}}$.

[5]Roughly speaking, Geißer *et al.* [2022] define a multi-valued heuristic H_1 to be more informed than another heuristic H_2 if using H_1 results in less node expansions than when using H_2.

To summarize, using multi-valued heuristics in the context of multi-objective shortest-path problems offers the potential to dramatically speed up multi-objective search algorithms. However, it also raises two key challenges:

C1 How to efficiently compute informative multi-value heuristics?

C2 How to effectively use multi-value heuristics to prune nodes in heuristic search algorithms?

5.2 Correlated Objectives

In many multi-objective settings there is some correlation between the different objectives which may be positive or negative. For example, in road networks, one may consider distance and time as two positively-correlated objectives (shorter routes are typically fast to traverse). Alternatively, one may consider distance and how beautiful a route is as two negatively-correlated objectives (scenic routes are typically longer to traverse).

Clearly, the correlation between the objectives may have a dramatic effect on the size and structure of Π^*. In the extreme case, where all objectives are identical (and hence perfectly correlated), the multi-objective problem essentially collapses to a single-objective problem. When the correlation is high, $|\Pi^*|$ may be extremely large but $|\Pi^*_\varepsilon|$ may be quite small, even for small approximation factors. However, it is not clear how correlation is defined in our setting: is this correlation along edges? along paths? Moreover, it is not clear how correlation can be used to design efficient algorithms.

Interestingly, these types of questions received little attention from the research community and, to the extent of our knowledge, these were primarily focused on the bi-objective setting. Mote *et al.* [1991] evaluated random graphs and grid graphs with a positive correlation between the two objectives associated with each edge. They showed that $|\Pi^*|$ decreases when the correlation increases and that in such cases $|\Pi^*|$ is relatively small. Similar results were shown by Brumbaugh-Smith and Shier [1989] when evaluating graphs where the two objectives associated with each edge are randomly generated from a bi-variate normal distribution.

The only work that introduces a formal model related to correlated objectives is the work of Müller-Hannemann and Weihe [2006]. Specifically, their model can be used to describe (to some extent) the structure of road networks, communication networks and train graphs.[6] Roughly speaking, this model assumes that edges can be classified into a small number of classes (e.g., in road networks, edges can be classified into highways, national roads, local roads etc.). Consequently, solutions on Π^* often have the structure that edge class increase and then decrease (such paths are termed *bitonic*). For example, it is quite atypical that solutions in a road network that belong to Π^* will include a local road connecting two highways or that in a train network paths belonging to Π^* include a local train between two high-speed trains. While in the worst-case graphs that fall under this model may include an exponentially large Pareto-optimal solution set, the empirical evaluation of Müller-Hannemann and Weihe suggest that the size of Π^* for each visited state can be regarded as constant for all practical purposes.

[6] For a precise definition of train graphs, see [Schulz *et al.*, 2000].

Instance	$\lvert\Pi^*\rvert$	D-T-M	T-D-M	M-T-D
1	4.4k	10.9	3.7	1.5
2	26.6k	117.1	35.9	21.4
3	41.7k	1,068.0	205.5	94.9
4	108.9k	2,546.7	783.2	472.8

Table 1: Runtime (in seconds) of a variant of EMOA* for different lexicographic ordering of cost functions on four instances.

A deeper understanding of how objectives are correlated can, and should, be used in practical applications. To this end, we suggest to explore the following research challenges:

C3 How to define the correlation between different objectives?

C4 Assuming we know that the objectives are correlated, how can this be used to speed up heuristic search algorithms?

5.3 Ordering Objectives

Recall that in multi-objective search problem it is common practice to order nodes in lexicographical order. This implies that there is some intrinsic ordering between the different objectives. Interestingly, this order may have a dramatic effect on the algorithm's running times. For example, we evaluated (Tbl. 1) the running time of LazyLTMOA* [Hernandez *et al.*, 2023b], a variant of EMOA*, on four tri-objective instances of the New York (NY) map from the `9'th DIMACS Implementation Challenge: Shortest Path`. Here, the first, second and third objectives are the travel time T, the travel distance D and the economic cost M (see [Pulido *et al.*, 2015] for a precise definition of the economic cost).

As expected, the runtime increases when $|\Pi^*|$ grows. However, we observed that order M-T-D can be several times faster (two orders of magnitude faster in instance three) than order D-T-M. A similar phenomena was observed for two objectives [Hernández *et al.*, 2023a; Ahmadi *et al.*, 2021].

Thus, we suggest to explore the following challenge:

C5 How to choose the best lexicographic order when ordering nodes to expand?

5.4 Paralelizing Multi-Objective Algorithms

In early work, Guerriero and Musmanno [2001] state that "parallel computing [...] represents the main goal for future developments". However, there is surprisingly little work on parallel multi-objective shortest-path algorithms.

More than a decade later, Sanders and Mandow [2013] present a parallel variant of one of the first algorithms for the bi-objective setting [Martins, 1984].Their approach, focused on the bi-objective setting with an emphasis on the

theoretical asymptotic running time, targets the heap operations used to order OPEN as the main source of parallelism. However, no experiments are provided and the authors state that their algorithm "might be too complicated to be practical". Soon after, Erb *et al.* [2014] present a parallel bi-objective shortest-path algorithm that uses weight-balanced B-trees with bulk updates. Their work, which was empirically evaluated and demonstrated dramatic speedups builds on the the work by Sanders and Mandow [2013] that on first glance looks impractical. Finally, Medrano and Church [2015] present another parallel approach for computing the set of extreme solutions for the bi-objective setting. Their results indicate the applicability of their method both on small-scale personal machines as well as large-scale shared memory supercomputers.

Additional work is a modification of BOA*, suggested by Ahmadi *et al.* [2021], in which one search runs from the source and another from the target. These searches may be executed in parallel with limited information reuse but no big empirical improvement was reported when paralelization was used. de las Casas *et al.* later adopts the similar modification for their bi-objective [de las Casas *et al.*, 2021b] and multi-objective [de las Casas *et al.*, 2021c] search algorithms.

To summarize, parallelizing multi-objective search is extremely under-explored and a challenge, already identified over twenty years ago, largely left unnoticed is

C6 How to efficiently and effectively parallelize search algorithms in the multi-objective setting?

5.5 Memory Footprint

Multi-Objective search algorithms often suffer from a large memory footprint, especially when the number of solutions in Π^* and the number of states S are large. This is because we typically store multiple paths for each state that need to be allocated in memory. For example, Ahmadi *et al.* [2021] reports the use of 96GB of memory (allocating 1B search states). While runtime is key in multi-objective search performance, typically improving memory usage implies increasing runtime [Coego *et al.*, 2009; Kothare *et al.*, 2022].

To cope with high memory usage, Kothare *et al.* [2022] proposed PE-EMOA* which applies the notion of partial expansion (see, e.g., [Yoshizumi *et al.*, 2000]) to the multi-objective setting. PE-EMOA* uses a user-defined hyperparameter to balance runtime and memory footprint and was shown to be effective in grid maps with a large branching factor [Rivera *et al.*, 2020]. However in road maps, the improvement in memory usage is negligible.

Another recent technique to save memory was presented by Ahmadi *et al.* [2021]. Considering that BOA* only expands nodes once, they propose to recycle the memory used to store already-processed nodes, while storing their backtracking information in other compact data structures. They report a decrease of $5\times$ in memory used on average on road maps while also slightly improving the runtime. Other techniques to save memory include extensions of linear-memory algorithms for single objective search to the multi-objective setting [Coego *et al.*, 2009; Coego *et al.*, 2013].

C7 How to efficiently cope with the memory footprint in heuristic-search algorithms without compromising on runtime?

6 Benchmarks

Key to the advancement of a research field that contains the empirical evaluation of algorithms is a standard set of benchmarks that spans multiple domains, each with its own characteristics. For example, the standardization of such a set of benchmarks in the MAPF domain presented by Stern *et al.* [2019] played a critical role in the advancement of the field. Similarly, the International Planning Competition (IPC) (see, e.g. [Vallati and others, 2015]) has been an important driver for research in the field of classical planning. In particular, they have resulted in a language for describing planning domains and problems (PDDL), a body of benchmark domains and problems in that language, and the ability to directly compare different generative planning techniques. Unfortunately, the multi-objective community is lacking such a set of benchmarks with different papers using different benchmarks to evaluate their algorithms. In this section, we review different benchmarks used in the hope that this will spark a discussion by the community which, in turn, will lead to a standard set of benchmarks.

Many papers (see, e.g., [Hernández *et al.*, 2023a; Pulido *et al.*, 2015]) use the `9'th DIMACS Implementation Challenge: Shortest Path`[7] which is a set of road networks with two cost functions available. A third cost function was suggested by Enriqueand and Lawrence [2011] and used in subsequent studies. A recent paper by Weise and Mostaghim [2022] proposes five objective functions commonly-available on road networks representing distance, traveling time, delays caused by accidents, curvature and elevation. These can be applied to any real-world data such as `OpenStreetMap`.[8]

Another set of benchmarks used (see, e.g., [de las Casas *et al.*, 2021c; de las Casas *et al.*, 2021b]) is `NetMaker`[9] which allows one to generate synthetic graphs with a moderate number of nodes. Alternative studies (see, e.g., [Pulido *et al.*, 2015; Ren *et al.*, 2022c]) suggested to use four-connected grids, where each component of the cost vector of an edge is a randomly generated integer from a given range.

Recently, Geißer *et al.* [2022] show that some well-known planning domains can be seen as multi-objective. For example, in the Sokoban puzzle, the number of moves and the number of pushes are two different objectives.

C8 How to generate a publicly-available, diverse, comprehensive set of benchmarks that will be used by the entire community?

[7] http://www.diag.uniroma1.it//~challenge9/.
[8] https://www.openstreetmap.org/.
[9] https://www.netmaker.org/

7 Summary and Discussion

Multi-objective search is an important research topic with applications ranging from route planning for power lines considering economic and ecological impacts [Bachmann *et al.*, 2018], transporting hazardous materials considering travel distance and risk [Bronfman *et al.*, 2015], and inspecting a region of interest using cameras placed on-board robotic platforms [Fu *et al.*, 2019; Fu *et al.*, 2021].

While being an active area of research for several decades, there is ample room to improve the state-of-the-art by addressing the key challenges that we described in this paper. A catalyst that will allow the field to further progress is a unified suite of benchmarks agreed upon by the community. To this end, we lay the groundwork by reviewing different domains used in different papers. However, the work is not complete: a comprehensive set of benchmarks should be diverse and analyzed according to how much the objectives are correlated and what the structure of the Pareto-optimal solution set for each benchmark is (see Sec. 5).

Acknowledgements

This research was supported by the United States-Israel Binational Science Foundation (BSF) grants no. 2019703 and 2021643 and by the Israeli Ministry of Science & Technology grants No. 3-16079 and 3-17385. The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant number 2121028. The research at Universidad San Sebastián was supported by the National Center for Artificial Intelligence CENIA FB210017, Basal ANID, and the Centro Ciencia & Vida, FB210008, Financiamiento Basal ANID.

References

[Ahmadi *et al.*, 2021] S. Ahmadi, G. Tack, D. Harabor, and P. Kilby. Bi-objective search with bi-directional A*. In *ESA*, volume 204, pages 3:1–3:15, 2021.

[Altwaijry and Menai, 2012] Najwa Altwaijry and Mohamed El-bachir Menai. Data structures in multi-objective evolutionary algorithms. *J. Comput. Sci. Technol.*, 27(6):1197–1210, 2012.

[Bachmann *et al.*, 2018] D. Bachmann, F. Bökler, J. Kopec, K. Popp, B. Schwarze, and F. Weichert. Multi-objective optimisation based planning of power-line grid expansions. *ISPRS I. J. of Geo-Inf.*, 7(7):258, 2018.

[Breugem *et al.*, 2017] T. Breugem, T. Dollevoet, and W. van den Heuvel. Analysis of FPTASes for the multi-objective shortest path problem. *Comput. Oper. Res.*, 78:44–58, 2017.

[Bronfman *et al.*, 2015] A. Bronfman, V. Marianov, G. Paredes-Belmar, and A. Lüer-Villagra. The maximin HAZMAT routing problem. *Eur. J. Oper. Res.*, 241(1):15–27, 2015.

[Brumbaugh-Smith and Shier, 1989] J. Brumbaugh-Smith and D. Shier. An empirical investigation of some bicriterion shortest path algorithms. *Eur. J. Oper. Res.*, 43(2):216–224, 1989.

[Clímaco and Pascoal, 2012] J. Clímaco and M. Pascoal. Multicriteria path and tree problems: discussion on exact algorithms and applications. *Int. Trans. Oper. Res.*, 19(1-2):63–98, 2012.

[Coego *et al.*, 2009] J. Coego, L. Mandow, and J. L. P. de la Cruz. A new approach to iterative deepening multiobjective A*. In *AI*IA*, pages 264–273, 2009.

[Coego *et al.*, 2013] J. Coego, L. Mandow, and J. L. P. de la Cruz. A comparison of multiobjective depth-first algorithms. *J. Intell. Manuf.*, 24:821–829, 2013.

[Current and Marsh, 1993] J. Current and M. Marsh. Multiobjective transportation network design and routing problems: Taxonomy and annotation. *Eur. J. Oper. Res.*, 65(1):4–19, 1993.

[de las Casas *et al.*, 2021a] P. Maristany de las Casas, R. Borndörfer, L. Kraus, and A. Sedeño-Noda. An FPTAS for dynamic multiobjective shortest path problems. *Algorithms*, 14(2):43, 2021.

[de las Casas *et al.*, 2021b] P. Maristany de las Casas, L. Kraus, A. Sedeño-Noda, and R. Borndörfer. Targeted multiobjective dijkstra algorithm. *CoRR*, abs/2110.10978, 2021.

[de las Casas *et al.*, 2021c] P. Maristany de las Casas, A. Sedeño-Noda, and R. Borndörfer. An improved multiobjective shortest path algorithm. *Comput. Oper. Res.*, 135:105424, 2021.

[Ehrgott, 2005] M. Ehrgott. *Multicriteria Optimization (2nd ed.)*. Springer, 2005.

[Emmerich and Deutz, 2018] M. Emmerich and A. Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing*, 17(3):585–609, 2018.

[Enriqueand and Lawrence, 2011] M. Enriqueand and M. Lawrence. Multiobjective route planning with precalculated heuristics. In *Portuguese Conf. on AI*, pages 98–107, 2011.

[Erb *et al.*, 2014] S. Erb, M. Kobitzsch, and P. Sanders. Parallel bi-objective shortest paths using weight-balanced b-trees with bulk updates. In *SEA*, pages 111–122, 2014.

[Fieldsend, 2020] Jonathan E. Fieldsend. Data structures for non-dominated sets: implementations and empirical assessment of two decades of advances. In *Genetic and Evolutionary Computation Conference*, pages 489–497. ACM, 2020.

[Fu *et al.*, 2019] M. Fu, A. Kuntz, O. Salzman, and R. Alterovitz. Toward asymptotically-optimal inspection planning via efficient near-optimal graph search. In *RSS*, 2019.

[Fu *et al.*, 2021] M. Fu, O. Salzman, and R. Alterovitz. Computationally-efficient roadmap-based inspection planning via incremental lazy search. In *ICRA*, pages 7449–7456, 2021.

[Geißer *et al.*, 2022] F. Geißer, P. Haslum, S. Thiébaux, and F. Trevizan. Admissible heuristics for multi-objective planning. In *ICAPS*, pages 100–109, 2022.

[Goldin and Salzman, 2021] B. Goldin and O. Salzman. Approximate bi-criteria search by efficient representation of subsets of the Pareto-optimal frontier. In *ICAPS*, pages 149–158, 2021.

[Guerriero and Musmanno, 2001] F Guerriero and R Musmanno. Label correcting methods to solve multicriteria shortest path problems. *J. Opt. Theory App.*, 111(3):589–613, 2001.

[Halffmann *et al.*, 2022] P. Halffmann, L. Schäfer, K. Dächert, K. Klamroth, and S. Ruzika. Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. *Journal of Multi-Criteria Decision Analysis*, 29(5-6):341–363, 2022.

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

[Hart *et al.*, 1968] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[Haslum and Geffner, 2000] P. Haslum and H. Geffner. Admissible heuristics for optimal planning. In *International Conference on Artificial Intelligence Planning Systems*, pages 140–149, 2000.

[Hayes and others, 2022] C. F Hayes et al. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agent. Multi Agent Syst.*, 36(1):1–59, 2022.

[Hernández *et al.*, 2023a] C. Hernández, W. Yeoh, J. A. Baier, H. Zhang, L. Suazo, S. Koenig, and O. Salzman. Simple and efficient bi-objective search algorithms via fast dominance checks. *Artif. Intell.*, 314:103807, 2023.

[Hernandez *et al.*, 2023b] C. U. Hernandez, H. Zhang, O. Salzman, J. A. Baier, W. Yeo, A. Felner, S. Kumar, and S. Koenig. Multi-objective search via lazy and efficient dominance checks. In *IJCAI*, 2023.

[Hwang and Masud, 2012] C-L Hwang and A. S. Masud. *Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media, 2012.

[Jaszkiewicz and Lust, 2018] Andrzej Jaszkiewicz and Thibaut Lust. ND-tree-based update: A fast algorithm for the dynamic nondominance problem. *IEEE Trans. Evol. Comput.*, 22(5):778–791, 2018.

[Koenig and Likhachev, 2002] S. Koenig and M. Likhachev. D*-lite. In *AAAI*, pages 476–483, 2002.

[Kothare *et al.*, 2022] V. Kothare, Z. Ren, S. Rathinam, and H. Choset. Enhanced multi-objective A* with partial expansion. *arXiv:2212.03712*, 2022.

[Legriel *et al.*, 2010] J. Legriel, C. Le Guernic, S. Cotton, and O. Maler. Approximating the Pareto front of multi-criteria optimization problems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 69–83, 2010.

[Li *et al.*, 2015] B. Li, J. Li, K. Tang, and X. Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):1–35, 2015.

[Mandow and De La Cruz, 2005] L. Mandow and J. L. P. De La Cruz. A new approach to multiobjective A* search. In *IJCAI*, pages 218–223, 2005.

[Mandow and De La Cruz, 2010] L. Mandow and J. L. P. De La Cruz. Multiobjective A* search with consistent heuristics. *J. ACM*, 57(5):1–25, 2010.

[Martins, 1984] E. Q. V. Martins. On a special class of bi-criterion path problems. *Eur. J. Oper. Res.*, 17(1):85–94, 1984.

[Medrano and Church, 2015] F. A. Medrano and R. L. Church. A parallel computing framework for finding the supported solutions to a biobjective network optimization problem. *J. Multi-Criteria Decis. Anal.*, 22(5-6):244–259, 2015.

[Miettinen, 2012] K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

[Mote *et al.*, 1991] J. Mote, I. Murthy, and D. L Olson. A parametric approach to solving bicriterion shortest path problems. *Eur. J. Oper. Res.*, 53(1):81–92, 1991.

[Müller-Hannemann and Weihe, 2006] M. Müller-Hannemann and K. Weihe. On the cardinality of the Pareto set in bicriteria shortest path problems. *Ann. Oper. Res.*, 147(1):269–286, 2006.

[Pangilinan and Janssens, 2007] J. M. Pangilinan and G. Janssens. Evolutionary algorithms for the multiobjective shortest path problem. *World Academy of Science, Engineering and Technology*, 25:205–210, 2007.

[Papadimitriou and Yannakakis, 2000] C. H Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FoCS*, pages 86–92, 2000.

[Perny and Spanjaard, 2008] P. Perny and O. Spanjaard. Near admissible algorithms for multiobjective search. In *ECAI*, pages 490–494, 2008.

[Pommerening *et al.*, 2014] F. Pommerening, G. Röger, M. Helmert, and B. Bonet. LP-based heuristics for cost-optimal planning. In *ICAPS*, 2014.

[Pulido *et al.*, 2015] F. J. Pulido, L. Mandow, and J. L. P. De la Cruz. Dimensionality reduction in multiobjective shortest path search. *Comput. Oper. Res.*, 64:60–70, 2015.

[Ren *et al.*, 2021] Z. Ren, S. Rathinam, and H. Choset. Multi-objective conflict-based search for multi-agent path finding. In *ICRA*, pages 8786–8791. IEEE, 2021.

[Ren *et al.*, 2022a] Z. Ren, S. Rathinam, M. Likhachev, and H. Choset. Multi-objective path-based D* lite. *IEEE Robotics Autom. Lett.*, 7(2):3318–3325, 2022.

[Ren *et al.*, 2022b] Z. Ren, S. Rathinam, M. Likhachev, and H. Choset. Multi-objective safe-interval path planning with dynamic obstacles. *IEEE Robotics Autom. Lett.*, 7(3):8154–8161, 2022.

[Ren *et al.*, 2022c] Z. Ren, R. Zhan, S. Rathinam, M. Likhachev, and H. Choset. Enhanced multi-objective A* using balanced binary search trees. In *SOCS*, pages 162–170, 2022.

[Rivera *et al.*, 2020] N. Rivera, C. Hernández, N Hormazábal, and J Baier. The 2^ k neighborhoods for grid path planning. *J. Artif. Intell. Res.*, 67:81–113, 2020.

[Rivera *et al.*, 2022] N. Rivera, J. A. Baier, and C. Hernández. Subset approximation of Pareto regions with bi-objective A*. In *AAAI*, pages 10345–10352, 2022.

[Roijers and Whiteson, 2017] D. M. Roijers and S. Whiteson. *Multi-Objective Decision Making*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2017.

[Roijers *et al.*, 2013] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.*, 48:67–113, 2013.

[Salzman and Stern, 2020] O. Salzman and R. Stern. Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems. In *AAMAS*, pages 1711–1715, 2020.

[Sanders and Mandow, 2013] P. Sanders and L. Mandow. Parallel label-setting multi-objective shortest path search. In *Int. Symp. on Parallel and Distributed Processing*, pages 215–224, 2013.

[Schulz *et al.*, 2000] F. Schulz, D. Wagner, and K. Weihe. Dijkstra's algorithm on-line: An empirical case study from public railroad transport. *J. Exp. Alg.*, 5:12–es, 2000.

[Sedeño-Noda and Colebrook, 2019] A. Sedeño-Noda and M. Colebrook. A biobjective dijkstra algorithm. *Eur. J. Oper. Res.*, 276(1):106–118, 2019.

[Sedeño-Noda and Raith, 2015] A. Sedeño-Noda and A. Raith. A dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem. *Comput. Oper. Res.*, 57:83–94, 2015.

[Serafini, 1987] P. Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In *Recent advances and historical development of vector optimization*, pages 222–232. Springer, 1987.

[Sievers and Helmert, 2021] S. Sievers and M. Helmert. Merge-and-shrink: A compositional theory of transformations of factored transition systems. *J. Artif. Intell. Res.*, 71:781–883, 2021.

[Skriver, 2000] A. J.V. Skriver. A classification of bicriterion shortest path (bsp) algorithms. *Asia Pacific J. of Oper. Res.*, 17(2):199–212, 2000.

[Skyler *et al.*, 2022] S. Skyler, D. Atzmon, A. Felner, O. Salzman, H. Zhang, S. Koenig, W. Yeoh, and C. Hernández. Bounded-cost bi-objective heuristic search. In *SOCS*, pages 239–243, 2022.

[Stern *et al.*, 2019] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, R. Barták, and E. Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *SOCS*, pages 151–159, 2019.

[Stewart and White III, 1991] B. Stewart and C. White III. Multiobjective A*. *J. ACM*, 38(4):775–814, 1991.

[Storandt, 2012] S. Storandt. Route planning for bicycles—exact constrained shortest paths made practical via contraction hierarchy. In *ICAPS*, pages 234–242, 2012.

[Tarapata, 2007] Z. M. Tarapata. Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms. *Int. J. Appl. Math*, 17(2), 2007.

[Tozer *et al.*, 2017] B. Tozer, T. Mazzuchi, and S. Sarkani. Many-objective stochastic path finding using reinforcement learning. *Expert Systems with Applications*, 72:371–382, 2017.

[Tsaggouris and Zaroliagis, 2009] G. Tsaggouris and C. D. Zaroliagis. Multiobjective optimization: Improved FPTAS for shortest paths and non-linear objectives with applications. *Theory of Com. Sys.*, 45(1):162–186, 2009.

[Ulungu and Teghem, 1991] EL Ulungu and J Teghem. Multi-objective shortest path problem: A survey. In *Workshop on Multicriteria Decision Making: Methods–Algorithms–Applications*, pages 176–188, 1991.

[Vallati and others, 2015] M. Vallati et al. The 2014 international planning competition: Progress and trends. *Ai Magazine*, 36(3):90–98, 2015.

[Vazirani, 2001] V. V. Vazirani. *Approximation algorithms*. Springer, 2001.

[Vincke, 1976] P. Vincke. A new approach to multiple criteria decision-making. In *Multiple Criteria Decision Making*, pages 341–350, 1976.

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

[Weise and Mostaghim, 2022] J. Weise and S. Mostaghim. A scalable many-objective pathfinding benchmark suite. *IEEE Trans. Evol. Comput.*, 26(1):188–194, 2022.

[Yoshizumi *et al.*, 2000] T. Yoshizumi, T. Miura, and T. Ishida. A* with partial expansion for large branching factor problems. In *AAAI*, pages 923–929, 2000.

[Zhang *et al.*, 2016] Y. Zhang, Z.-J M. Shen, and S. Song. Parametric search for the bi-attribute concave shortest path problem. *Transportation Research Part B: Methodological*, 94:150–168, 2016.

[Zhang *et al.*, 2022a] H. Zhang, O. Salzman, T. K. S. Kumar, A. Felner, C. Hernández, and S. Koenig. Anytime approximate bi-objective search. In *SOCS*, pages 199–207, 2022.

[Zhang *et al.*, 2022b] H. Zhang, O. Salzman, TK S. Kumar, A. Felner, C. Hernández, and S. Koenig. A*pex: Efficient approximate multi-objective search on graphs. In *ICAPS*, pages 394–403, 2022.