Journal of Artificial Intelligence Research 48 (2013) 717-732 Submitted 6/13; published 11/13

Research Note

A Case of Pathology in Multiobjective Heuristic Search

José Luis Pérez de la Cruz PEREZ@LCC.UMA.ES
Lawrence Mandow LAWRENCE@LCC.UMA.ES
Enrique Machuca MACHUCA@LCC.UMA.ES
Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga
Bulevar Louis Pasteur, 35. Campus de Teatinos, 29071
Málaga (Spain)

Abstract

This article considers the performance of the MOA* multiobjective search algorithm with heuristic information. It is shown that in certain cases blind search can be more efficient than perfectly informed search, in terms of both node and label expansions.

A class of simple graph search problems is defined for which the number of nodes grows linearly with problem size and the number of nondominated labels grows quadratically. It is proved that for these problems the number of node expansions performed by blind MOA* grows linearly with problem size, while the number of such expansions performed with a perfectly informed heuristic grows quadratically. It is also proved that the number of label expansions grows quadratically in the blind case and cubically in the informed case.

1. Introduction

Heuristic search algorithms are central to problem solving in Artificial Intelligence and to many practical applications in Operations Research. In heuristic search some additional information is provided to the algorithm with the aim of reducing the computational effort needed to find a solution.

However, sometimes this goal is not achieved. On the contrary, in certain cases it has been shown that the use of better heuristics implies a worsening of performance. For example, a well-known fact arising in some bipersonal games is that of *lookahead pathology*, that is, that the deeper is the exploration performed (and hence the better is suppossedly the heuristic minimax value assigned to the position), the worse is the decision taken (Nau, 1982). Recently, the same phenomenon was described in one-agent real-time search (Lustrek & Bulitko, 2008; Nau, Lustrek, Parker, Bratko, & Gams, 2010).

Even with statically precomputed heuristics some pathologies have been found. For instance, let us consider the standard A* algorithm (Hart, Nilsson, & Raphael, 1968). It is known that the algorithm is admissible when provided with optimistic heuristic cost estimates and that, when these estimates are also consistent, more informed heuristics always result in equally or more efficient search (see Pearl, 1984, especially pp. 75–85). However, when the heuristic is optimistic but not consistent, algorithm A* can perform $O(2^n)$ node expansions when the search is performed on a graph with n nodes and arc costs are not bounded (Martelli, 1977). Notice that if no heuristic is used, then A* performs like Dijkstra's algorithm and can never exhibit such exponential performance.

This paper deals with a pathology arising in certain extension of A* to multiobjective search problems. In decision making situations where more than one criterion is involved, the concept of optimal solution is frequently replaced by Pareto optimality, that is, a solution is better than other if it improves with respect to at least one criterion without worsening the others (Ehrgott, 2005). Since Pareto optimality is a partial order relation, solving these problems usually results in a set of Pareto-optimal solutions, that represent the optimal trade-offs between the criteria being optimized. The importance of research in multiobjective search algorithms is two-fold. In the first place, many graph search problems can benefit directly from multiobjective analysis (De Luca Cardillo & Fortuna, 2000; Gabrel & Vanderpooten, 2002; Refanidis & Vlahavas, 2003; Müller-Hannemann & Weihe, 2006; Dell'Olmo, Gentili, & Scozzari, 2005; Ziebart, Dey, & Bagnell, 2008; Wu, Campbell, & Merz, 2009; Delling & Wagner, 2009; Fave, Canu, Iocchi, Nardi, & Ziparo, 2009; Mouratidis, Lin, & Yiu, 2010; Caramia, Giordani, & Iovanella, 2010; Boxnick, Klöpfer, Romaus, & Klöpper, 2010; Klöpper, Ishikawa, & Honiden, 2010; Wu, Campbell, & Merz, 2011; Machuca & Mandow, 2011). On the other hand, other multicriteria preference models used in graph search typically look for a subset of Pareto-optimal solutions (Mandow & Pérez de la Cruz, 2003; Perny & Spanjaard, 2005; Galand & Perny, 2006; Galand & Spanjaard, 2007; Galand, Perny, & Spanjaard, 2010). Therefore, improvements in performance of multiobjective algorithms can guide the development of efficient algorithms for other multicriteria decision rules.

Two direct extensions of A* that accept general (multiobjective) heuristic functions have been proposed in the literature: MOA* (Stewart & White, 1991) and NAMOA* (Mandow & Pérez de la Cruz, 2005). NAMOA* uses label selection to guide the exploration. From a formal point of view, a recent analysis (Mandow & Pérez de la Cruz, 2010a) has shown that the algorithm is admissible with optimistic heuristics, and that its efficiency, measured by the number of label expansions, improves with more informed consistent heuristics. Furthermore, the number of such expansions is optimal with respect to a class of admissible algorithms. In other words, NAMOA* inherits the beneficial properties of A*.

MOA* uses node selection (as opposed to label selection) to guide the exploration, and is also known to be admissible with optimistic heuristics (Stewart & White, 1991). The development of MOA* prompted a number of related formal developments and extensions (Dasgupta, Chakrabarti, & DeSarkar, 1995, 1999; Perny & Spanjaard, 2002; Mandow & Pérez de la Cruz, 2003; Perny & Spanjaard, 2005), and is still cited as an algorithm of choice in recent applications (De Luca Cardillo & Fortuna, 2000; Fave et al., 2009; Klöpper et al., 2010). A previous formal analysis showed that there exist problems for which blind MOA* performs $\Theta(2^n)$ node expansions on graphs with n nodes (Mandow & Pérez de la Cruz, 2010b). However, the formal analysis of MOA* remained incomplete. In particular, the efficiency of the algorithm was never related to the precision of consistent heuristics. A recent empirical analysis (Machuca, Mandow, Pérez de la Cruz, & Ruiz-Sepúlveda, 2010) has shown that, in certain cases, MOA* performs much worse than NAMOA* over biobjective random problems with different correlations. Quite surprisingly, this analysis has also revealed that heuristic MOA* actually performs consistently worse than uninformed MOA*.

This paper is part of an investigation into the formal properties of MOA* and NAMOA*. We formally show that the performance of MOA*, measured in terms of the number of label or node expansions, does not improve in general with more informed heuristics. More precisely, we define a class of simple graph search problems and prove that the use of perfect heuristic information in MOA* yields more computational effort than the use of no heuristic information. In other words, blind MOA* is in these cases more efficient than perfectly informed MOA*, in terms of both node and label expansions.

The article is organized as follows. First, some necessary concepts are presented and algorithm MOA* is briefly described (Section 2). Then in Section 3 a class of simple multiobjective search problems is defined. The performance of MOA* over this class of problems is analyzed for blind and perfectly informed cases in Sections 4 and 5 in terms of node expansions, and in Section 6 in terms of label expansions. Finally, some conclusions and future work are described.

2. Background

In multiobjective decision problems each alternative is evaluated according to a set of q different objectives usually grouped in a vector $\vec{y} = (y_1, y_2, \ldots y_q), \vec{y} \in \mathbb{R}^q$. Preference between vectors $\vec{x}, \vec{y}$ is defined by the so-called *Pareto order* or *dominance relation* ($\prec$) as follows: $\vec{x} \prec \vec{y}$ if and only if for all objectives i it holds that $x_i \leq y_i$ and at least for an objective j it holds that $x_j < y_j$. Given a set of vectors Y, the subset of nondominated vectors $\mathrm{nd}(Y)$ in Y is defined as $\mathrm{nd}(Y) = \{\vec{y} \in Y \mid \nexists \vec{x} \in Y \quad \vec{x} \prec \vec{y}\}$.

The solution to a multiobjective problem consists of the set of *Pareto-optimal* or *non-dominated* solutions, that is, the set of solutions such that their costs are nondominated in the set of solution costs.

In a multiobjective graph search problem, a single source and a set of destination nodes are designated in a given graph $G = (N, A)$. Pairs of nodes $n, n' \in N$ may be joined by directed arcs $(n, n') \in A$ labelled with vector costs $\vec{c}(n, n') \in \mathbb{R}^q$. A path P in the graph is any sequence of nodes joined by consecutive arcs and the cost $\vec{c}(P)$ of P is the sum of the costs of its component arcs. The solution to the problem is the set of all paths P joining source and destination nodes and such that $\vec{c}(P)$ is nondominated in the set of solution costs.

2.1 MOA* Algorithm

MOA* is a well-known algorithm that performs multiobjective heuristic graph search (Stewart & White, 1991). Its pseudocode (slightly adapted from the original: Stewart & White, 1991) is presented in Table 1. MOA* presents many similarities with A*. Two sets of nodes $OPEN$ and $CLOSED$ are used to control the search. Initially, the source node is the only open node. Newly generated nodes create a pointer to their parents. However, MOA* does not construct a search tree like A*, but rather an acyclic directed graph. This is due to the fact that each node may be reached by several optimal (nondominated) paths. The scalar cost functions g, h, f are generalized to functions G, H, F that return sets of vectors for each node. Additionally, the $\mathrm{LABEL}(n', n)$ sets keep the subsets of vectors in $G(n')$ that arise from paths to n' coming from n.

Function $G(n)$ refers to the set of nondominated cost vectors among all paths already found to n. The heuristic function $H(n)$ returns also a set of vectors, estimating the costs of

1. INITIALIZE a set $OPEN$ with the start node s, and empty sets, $SOLN$, C, $CLOSED$ and $LABEL$.
2. CALCULATE the set ND of nodes n in $OPEN$ such that at least one estimate $\vec{f} \in F(n)$ is not dominated by the estimates of other open nodes or by any solution cost of C.
3. If ND is empty, then

   —Terminate returning the set of solution paths that reach nodes in $SOLN$ with costs in C.

   else

   —Choose a node n from ND using a domain-specific heuristic, breaking ties in favour of goal nodes, and move n from $OPEN$ to $CLOSED$.
4. Do bookkeeping to maintain accrued costs and node selection function values.
5. IDENTIFY SOLUTIONS. If n is solution node, then

   —Include n in $SOLN$ and its current costs into C.

   —Remove dominated costs from C.

   —Go back to step 2.
6. EXPAND n and examine its successors. For all successors nodes m of n do:
   (a) If m is a newly generated node, then
      i. Establish a pointer from m to n.
      ii. Set $G(m) = LABEL(m, n)$.
      iii. Compute $F(m)$.
      iv. Add m to $OPEN$.

   (b) Otherwise, m is not new, so do the following,
      i. If any potentially nondominated paths to m have been discovered, then, for each one, do the following.
      —Ensure that its cost is in $LABEL(m, n)$, and therefore in $G(m)$.
      —If a new cost was added to $G(m)$ then, purge from $LABEL(m, n)$ dominated costs, and if m was in $CLOSED$, then move it to $OPEN$.
7. Go back to step 2.

Table 1: MOA* Algorithm.

all nondominated paths from n to destination nodes. The evaluation function $F(n)$ returns a set of cost estimates for n, $F(n) = \text{nd}\{\vec{g} + \vec{h} \mid \vec{g} \in G(n) \wedge \vec{h} \in H(n)\}$.

Later on it will be useful to define $H^*(n)$ as the function that returns the set of costs of all actual nondominated paths from n to destinations nodes.

At each iteration MOA* computes *ND*, the subset of open nodes with a nondominated cost estimate, and selects a node from this subset. The admissibility of the algorithm does not depend on the particular selection procedure among nodes in *ND*. In the following, this additional selection procedure for nondominated nodes will be called *nd-selection rule*.

When a destination node is selected, it is added to *SOLN*, and its costs to *C*. Values of $F(n)$ dominated by vectors in *C* are never considered in *ND*. Search terminates when *ND* is empty, that is, all candidate nodes are dominated or have been explored.

The expansion of n generates all successors n' of n in the graph and adequate $G(n')$ values for them. If n' is new, then it is placed in *OPEN*, and the sets $G(n')$ and $\text{LABEL}(n', n)$ store the costs of all paths extended from n to n'. If n' is not new, then MOA* checks if a new nondominated value of $G(n')$ has been generated at the current step; if this is the case, $G(n')$ and LABEL(n', n) are properly updated, and, if n' was in $CLOSED$, it must be moved back again to *OPEN*.

Each pair $(n, \vec{g})$ such that n is a node and $\vec{g} \in G(n)$ is usually called a *label*. In MOA* all labels of a node n are expanded simultaneously once n is selected. Therefore, all labels reaching a single node at a given time are either simultaneously open or closed.

In the original paper (Stewart & White, 1991), some interesting properties of MOA* were proved. For example, it was proved that MOA* is *admissible* when $H(n)$ is optimistic.

Regarding comparison of admissible heuristics, a function $H(n)$ is defined to be *at least as informed* as another $H'(n)$ whenever for all $\vec{h}' \in H'(n)$ there exists some $\vec{h} \in H(n)$ such that $\vec{h}' \preceq \vec{h}$. In such case, it was proved that the set of nodes expanded by MOA* with H is a subset of those expanded with H' (theorem 4, p. 805). However, the authors recognized that nodes may be reopened even when the heuristic function is consistent, and hence that the set of expanded nodes is not a significant measure in the analysis of the performance of MOA*.

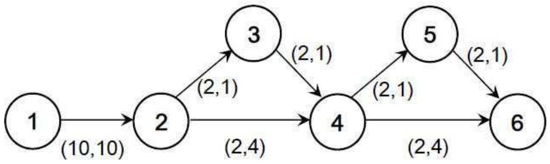

Figure 1: Graph $M(3, 10, 10, 2)$

3. A Class of Multiobjective Search Problems

For every $n \in \mathbb{N}$, let us consider problem graphs (Figure 1) with $2n$ nodes labeled $1, \ldots, 2n-1, 2n$ and with $3n-2$ arcs. For every even node $2i$ $(1 \leq i < n)$ there are outgoing arcs of form $(2i, 2i+1)$ and $(2i, 2i+2)$. For every odd node $2i+1$ $(1 \leq i < n-1)$ there is an outgoing arc of the form $(2i+1, 2i+2)$. There is also an arc $(1, 2)$. The start node is 1 and the goal node is $2n$. The cost of an arc $\vec{c}(i, j)$ is defined as follows: choose α to be either 2 or 4; then for every $i > 0$, $\vec{c}(2i, 2i+2) = (\alpha, 6-\alpha)$; and for every $i > 0$, $\vec{c}(2i, 2i+1) = \vec{c}(2i+1, 2i+2) = (3-\alpha/2, \alpha/2)$. In this way, we define only two possible sets of costs for the arcs, with the exception of $\vec{c}(1, 2) = (\kappa_1, \kappa_2)$ that is not subject to any restriction.

We shall refer to these problem graphs as *multiobjective chain graphs*. For every n, the corresponding set of multiobjective chain graphs will be denoted $\mathcal{M}_n$. We will denote as $M(n, \kappa_1, \kappa_2, \alpha)$ the graph in $\mathcal{M}_n$ such that $\vec{c}(1, 2) = (\kappa_1, \kappa_2)$ and $c(2i, 2i+2) = (\alpha, 6-\alpha)$. For example, Figure 1 shows $M(3, 10, 10, 2)$.

In a graph $M \in \mathcal{M}_n$ there are always 2^{n-1} different paths from the start to the goal node. In fact, to go from $2i$ to $2i+2$ you can choose either the simple path $< 2i, 2i+2 >$ (with cost $(\alpha, 6-\alpha)$) or the two-arc path $< 2i, 2i+1, 2i+2 >$ (with cost $(6-\alpha, \alpha)$) and there are $n-1$ independent choices like that. On the other hand, there are just n different path costs given by $\vec{c_n}^k = (\kappa_1, \kappa_2) + (2(n-1) + 2k, 4(n-1) - 2k)$, for every k such that

It. #	*OPEN*	$G(n) = F(n)$
1	$1 \leftarrow$	$(0, 0)$
2	$2 \leftarrow$	$(10, 10)$
3	$3 \leftarrow$	$(12, 11)$
	4	$(12, 14)$
4	$4 \leftarrow$	$(12, 14)(14, 12)$
5	$5 \leftarrow$	$(14, 15)(16, 13)$
	6	$(14, 18)(16, 16)$
6	$6 \leftarrow$	$(14, 18)(16, 16)(18, 14)$

Table 2: Trace of uninformed MOA* on $M(3, 10, 10, 2)$

$0 \leq k \leq n-1$. In an $M(n, \kappa_1, \kappa_2, 2)$ graph, the cost $\vec{c_n}^k$ corresponds to a path with $n-1-k$ arcs $(2i, 2i+2)$ and k paths $< 2i, 2i+1, 2i+2 >$. We will denote $C_n = \{\vec{c_n}^0, \ldots, \vec{c_n}^{n-1}\}$. Notice that for every solution cost $(y_1, y_2) \in C_n$ it holds that $y_1 + y_2 = \kappa_1 + \kappa_2 + 6(n-1)$, so all solution costs for a given M lie in a line with negative slope and hence none of these costs dominates another, that is, $\mathrm{nd}(C_n) = C_n$.

4. Blind MOA* on $\mathcal{M}_n$

A sample run of MOA* over $M(3, 10, 10, 2)$ (Figure 1) with $H(n) = \{\vec{0}\}$ (uninformed case) is provided in Table 2. Values of $G(n)$ include all nondominated costs from generated paths to the node. Since we are performing a blind search, values for $F(n)$ are the same as for $G(n)$.

We can observe in this trace that a node i is not selected until all nodes $j < i$ have been selected. That means that every node i, $1 \leq i \leq 2n-1$ is expanded once and just once. In this way, in our example MOA* performs exactly $2n$ node selections and $2n-1$ node expansions. This result is general and can be proved by induction on the number of iterations for every graph $M(n, \kappa_1, \kappa_2, \alpha)$.

Lemma 1 *If the input of MOA* is a graph $M(n, \kappa_1, \kappa_2, \alpha)$ and $\forall n\ H(n) = \{\vec{0}\}$, then MOA* performs exactly $2n-1$ node expansions.*

Proof. Let us consider the *OPEN* set at a certain iteration s ($s = 1, \ldots$) of the execution of MOA*. Let us call the *level* of s the integer $L = \lfloor s/2 \rfloor$. It is easily proved by induction on s that:

(i) at every odd iteration s of the algorithm ($s = 3, \ldots$), $OPEN = \{s, s+1\}$, the labels of s are $\{(a+3-\alpha/2, b+\alpha/2) \mid (a, b) \in C_L\}$; the labels of $s+1$ are $\{(a+\alpha, b+6-\alpha) \mid (a, b) \in C_L\}$; and the selected node is s.

(ii) At every even iteration s of the algorithm ($s = 4, \ldots$), $OPEN = \{s\}$, the labels of s are exactly C_L and the selected node is s.

From this follows that every node is selected exactly once and therefore the number of node expansions is exactly $2n-1$.

$\triangleleft$

n	$H^*(n)$
1	(14,18),(16,16),(18,14)
2	(4,8),(6,6),(8,4)
3	(4,5),(6,3)
4	(2,4),(4,2)
5	(2,1)
6	(0,0)

Table 3: Heuristic values for $M(3, 10, 10, 2)$

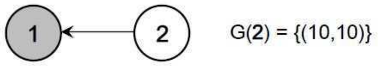

Figure 2: Search graph at iteration 2

5. Perfectly Informed MOA* on $\mathcal{M}_n$

A sample run of MOA* over $M(3, 10, 10, 2)$ (Figure 1) with $H(n) = H^*(n)$ (perfect information) is provided in Figures 2-9 and Table 4. Figures 2-9 show a trace of the search graph at each iteration. Closed nodes are shown in gray. Values of $G(n)$ are shown for each node in the Figures only when they are created, or change from the previous iteration.

In Table 4, for each iteration all nodes in *OPEN* are displayed and also their $G(n)$ values (that is, nondominated costs of paths generated from the start to the node). Values of $F(n)$ are computed adding to $G(n)$ values the estimations in Table 3. Since we are assuming perfect heuristic information, $F(n)$ values are always optimal solution costs, that is, nondominated costs of solution paths from node 1 to node 6. Then, for every iteration and every node n, $F(n) \subseteq C_3$.

Notice also that labels in C_3 (and hence in $F(n)$) are all nondominated, so in general there will be several open nondominated nodes. It is then necessary to provide an additional heuristic rule (step 3-else of the algorithm in Table 1) or *nd-selection rule*. In the example in Table 4, the following nd-selection rule is applied: select the node with the best lexicographic nondominated alternative (remember that the lexicographic order is a total order defined for the biobjective case as $(y_1, y_2) < (z_1, z_2)$ if and only if $y_1 < z_1$, or $y_1 = z_1$ and $y_2 < z_2$). It is also possible that the same nondominated cost appears in several open nodes. Then another procedure must be provided for breaking ties between open nodes with the same $\vec{f}$-label. It can be done, for instance, at random, or by selecting the newest node, or the oldest node in *OPEN* in a breadth-first fashion. The latter procedure has been followed in Table 4.

We can observe that the order of node selection in the example is

$$1 - 2 - 4 - 6 - 3 - 4 - 6 - 5 - 6$$

The pattern is: MOA* selects even nodes until the goal node is reached; then it selects an odd node $2i + 1$ and selects again all even nodes $2j, j > i$; and it is done again until every odd node has been selected once. In this way even nodes are in general selected several times. In our example node 4 is selected twice and node 6 is selected three times.

It#	$OPEN$	$G(n)$	$F(n)$
1	$1 \leftarrow$	$(0,0)$	$(\mathbf{14},\mathbf{18})(16,16)(18,14)$
2	$2 \leftarrow$	$(10,10)$	$(\mathbf{14},\mathbf{18})(16,16)(18,14)$
3	3	$(12,11)$	$(16,16)(18,14)$
	$4 \leftarrow$	$(12,14)$	$(\mathbf{14},\mathbf{18})(16,16)$
4	3	$(12,11)$	$(16,16)(18,14)$
	5	$(14,15)$	$(16,16)$
	$6 \leftarrow$	$(14,18)$	$(\mathbf{14},\mathbf{18})$
5	$3 \leftarrow$	$(12,11)$	$(\mathbf{16},\mathbf{16})(18,14)$
	5	$(14,15)$	$(16,16)$
6	$4 \leftarrow$	$(12,14)(14,12)$	$(\mathbf{14},\mathbf{18})(16,16)(18,14)$
	5	$(14,15)$	$(16,16)$
7	5	$(14,15)(16,13)$	$(16,16)(18,14)$
	$6 \leftarrow$	$(14,18)(16,16)$	$(\mathbf{14},\mathbf{18})(16,16)$
8	$5 \leftarrow$	$(14,15)(16,13)$	$(\mathbf{16},\mathbf{16})(18,14)$
9	$6 \leftarrow$	$(14,18)(16,16)(18,14))$	$(\mathbf{14},\mathbf{18})(16,16)(18,14)$

Table 4: Trace of perfectly informed MOA* on $M(3,10,10,2)$

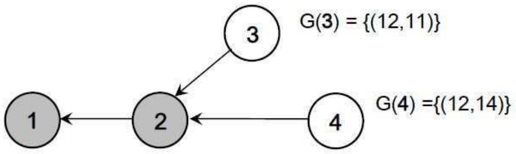

Figure 3: Search graph at iteration 3

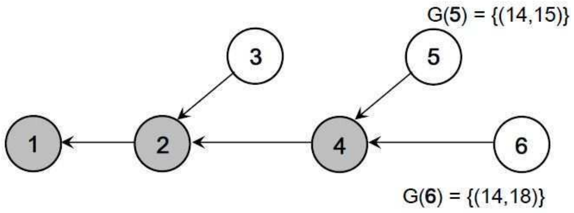

Figure 4: Search graph at iteration 4

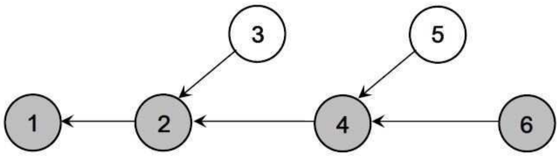

Figure 5: Search graph at iteration 5

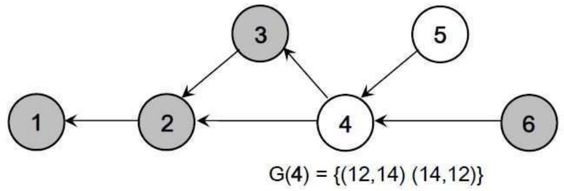

Figure 6: Search graph at iteration 6

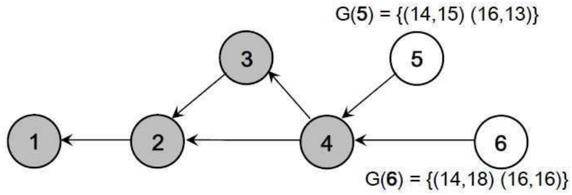

Figure 7: Search graph at iteration 7

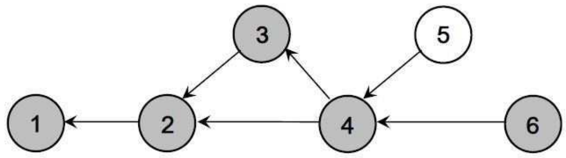

Figure 8: Search graph at iteration 8

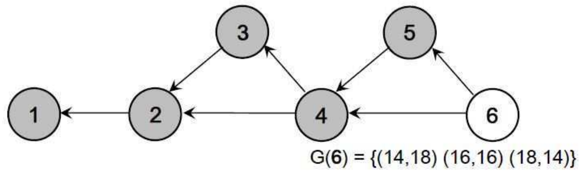

Figure 9: Search graph at iteration 9

The concrete order of expansion will depend on nd-selection and tie-breaking rules. However, we can prove some general results valid for any nd-selection rule that uses only heuristic $\vec{f}$ values (irrespective of the tie-breaking rule). Notice that nd-selection rules usually applied (as best lexicographic or best linear) are of this kind.

Lemma 2 *(i) Let $M(n, \kappa_1, \kappa_2, 2) \in \mathcal{M}_n$. Let their nondominated solution costs be $C = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,n-1}\}$. If the nd-selection rule is such that $\vec{c}^{\,0}$ is selected with preference to $\{\vec{c}^{\,1}, \ldots, \vec{c}^{\,n-1}\}$, then MOA* performs at least $n + \frac{n(n-1)}{2}$ node expansions.*

(ii) Analogously, let $M(n, \kappa_1, \kappa_2, 4) \in \mathcal{M}_n$. If the nd-selection rule is such that $\vec{c}^{\,n-1}$ is selected with preference to $\{\vec{c}^{\,0}, \ldots, \vec{c}^{\,n-2}\}$, then MOA performs at least $n + \frac{n(n-1)}{2}$ node expansions.*

Proof. We will prove part (i) (proof of part (ii) is entirely analogous). Let $M(n, \kappa_1, \kappa_2, 2) \in \mathcal{M}_n$. First remember that for each node j and each iteration s of the algorithm, the set $F(j)$ of estimated costs at j is a subset of $C_n = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,k}, \ldots, \vec{c}^{\,n-1}\} = \{(\kappa_1 + (2(n-1) + 2k, \kappa_2 + 4(n-1) - 2k)) \mid 0 \leq k \leq n-1\}$. Let us trace the first $n+1$ iterations of the algorithm. It can be shown that for every $i > 0$, $\vec{c}^{\,0}$ will appear in $F(2i)$, but $\vec{c}^{\,0}$ will never appear in $F(2i+1)$. Therefore, in the first iteration node 1 will be selected and expanded. Then all even nodes will be selected and expanded sequentially until node $2i$ is selected. That amounts to the first n expansions.

Elementary computations show that at this step, for every $0 < i \leq n$, $F(2i) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,n-i}\}$; for every $0 < i \leq n-1$, $F(2i+1) = \{\vec{c}^{\,1}, \ldots, \vec{c}^{\,n-i}\}$; and $OPEN = \{3, 5, \ldots, 2n-1\}$. Two observations can be made at this step: i) Let us consider odd nodes. Since for every odd node $2i+1$ $(0 < i < n)$ there is an optimal path going through it, all these open nodes must be selected and expanded before termination. That amounts to at least $n-1$ expansions, even if no reexpansion is assumed for such nodes; ii) Let us consider now nongoal even nodes $\{2, 4, \ldots, 2n-2\}$. At this iteration the number of labels associated to $2i$ is $n+1-i$. Since through every even node there exist n optimal costs, at the termination the number of labels for every even node must be exactly n, that is, there are $0+1+\ldots+(n-1) = \frac{n(n-1)}{2}$ labels for even nodes missing at this moment. If we prove that those labels are generated one at a time, that is, one in every expansion, we will have proved that at least another $\frac{n(n-1)}{2}$ expansions are needed.

Let us call an *episode* the subsequence of node expansions comprised between two consecutive odd node expansions (or between the last odd node expansion and the last expansion). In the example of Table 4, the episodes are $< 1, 2, 4, 6 >$, $< 3, 4, 6 >$ and $< 5, 6 >$. When an episode starts, there is no even node in *OPEN* (since an even node always has the nd-selected label $\vec{c}^{\,0}$, an odd node never has the nd-selected label $\vec{c}^{\,0}$, and an odd node has been selected). At the end of the episode, and by the same reason, there is again no even node in *OPEN.* Let us consider an episode $e = < 2j-1, 2j, 2j', \ldots, 2j^n >$. It is easily seen that the expansion of an even node $2j$ originates the opening of just an even node, namely $2(j+1)$, so the episode is always of the form $E = < 2j-1, 2j, 2(j+1), 2(j+2), \ldots, 2(j+m) >$. We will prove by induction on episodes that when every episode starts, for every even node $2i$ $(1 < i \leq n)$ there exist integers p, q such that: i) $F(2i) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,p}\}$; ii) $F(2i-2) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,q}\} = F(2i-1) \cup \{\vec{c}^{\,0}\}$; and iii) either $q = p$ or $q = p+1$. This is obviously true when the first episode finishes and the second episode starts. Assume it is true when episode e starts. We will show it remains true when it finishes, that is, when episode $e+1$ starts.

Consider the odd node $2j-1$ which starts the episode and assume $q = p$, that is, that $F(2j-1) = F(2j) - \{\vec{c}^{\,0}\}$. Then no new label is added to $F(2j)$, no reexpansion is needed and no modification is done, hence the stated relation among labels continues true. On the contrary, assume that $q = p+1$, that is, that $F(2j-1) = \{\vec{c}^{\,1}, \ldots, \vec{c}^{\,p+1}\}$, $F(2j-2) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,p+1}\}$, and $F(2j) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,p}\}$. Therefore one label $\vec{c}^{\,p+1}$ is added to $2j$. The stated relation between labels remain true for $2j-2$, $2j-1$ and $2j$ (only that we have now the other alternative $p = q$). However, $F(2j)$ has been modified and we must check the relation for $F(2j)$, $F(2j+1)$ and $F(2j+2)$. By hypothesis it was $F(2j+1) = \{\vec{c}^{\,1}, \ldots, \vec{c}^{\,p}\}$, and the expansion of $2j$ adds $\vec{c}^{\,p+1}$ to it, so also it becomes $F(2j) = F(2j+1)$. Concerning the relation between $F(2j)$ and $F(2j+2)$, if it was $F(2j+2) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,p}\}$, no further label is added to $F(2j+2)$, the relation holds (since $F(2j+2)$ has exactly one label less than $F(2j)$ now) and the episode finishes since no even node remains open. If it was $F(2j+2) = \{\vec{c}^{\,0}, \ldots, \vec{c}^{\,p-1}\}$, one label $\vec{c}^{\,p}$ is added to $F(2j+2)$ and the relation also becomes true, but $F(2j+2)$ has been modified. By induction on the length of the episode it could be proved that finally the relation holds for all modified nodes.

In any case, we see that each new added label immediately triggers the expansion of the node and labels are added to even nodes one at a time. Therefore, at least $\frac{n(n-1)}{2}$ expansions are needed to complete the labels of even nodes. The first expansion of the episode was one of an odd node, so it must not be computed; but the last expansion of the episode does not add any label, so we can compute $\frac{n(n-1)}{2}$ additional node expansions.

Now, adding togheter all performed expansions, we have at least $n + \frac{n(n-1)}{2}$ expansions, q. e. d.

◁

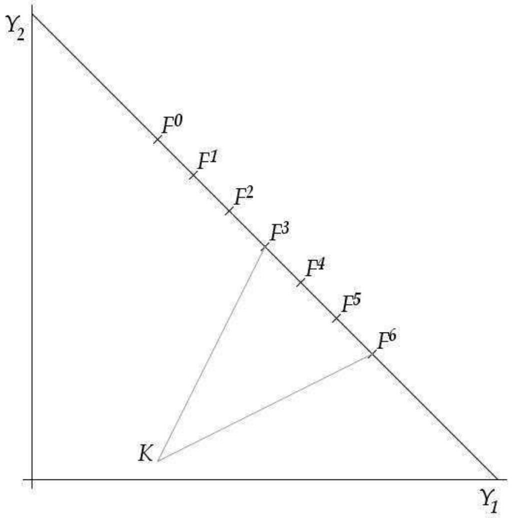

Figure 10: Construction of a $M(4, \kappa_1, \kappa_2, 2)$ graph for Lemma 3

Now we can prove the main lemma:

Lemma 3 *If the nd-selection rule depends only on $\vec{f}$ values, then for every $n \in \mathbb{N}$ there exists a graph $M(n, \kappa_1, \kappa_2, \alpha)$ such that when it is input to MOA* and $H(n) = H^*(n)$, then MOA* performs at least $n + \frac{n(n-1)}{2}$ node expansions.*

Proof. The idea of the proof is as follows: Lemma 2 asserts that if we order lexicographically the optimal costs of a graph $M(n, \kappa_1, \kappa_2, 2) \in \mathcal{M}_n$ and the selected one turns out to be the first one, then MOA* performs at least $n+\frac{n(n-1)}{2}$ node expansions (and analogously for every graph $M(n, \kappa_1, \kappa_2, 4) \in \mathcal{M}_n$ when the selected one is the last one). Then we will have Lemma 3 proved if for every n and every selection rule depending only on $\vec{f}$ values we show a graph $M(n, \kappa_1, \kappa_2, 2) \in \mathcal{M}_n$ satisfying that condition (or a graph $M(n, \kappa_1, \kappa_2, 4) \in \mathcal{M}_n$ satisfying the analogous condition).

Let $n \in \mathbb{N}$. Let us consider the line $y_1 + y_2 = 8(n-1) + 2$ and a sequence of $m = 2n - 1$ nondominated points on it, $(F^0, \ldots, F^{m-1})$ given by $F^i = (2(n-1)+2i+1, 6(n-1)-2i+1)$ (Figure 10 shows the line and the seven points $F^0, \ldots, F^6$ for $n = 4$). Any nd-selection rule will select one of them, say F^j, with preference over the others; in any case, for any F^j we can extract from $(F^0, \ldots, F^{m-1})$ a subsequence of n points $(F^j, F^{j+1}, \ldots, F^{j+n-1})$ or $(F^{j-n+1}, \ldots, F^{j-1}, F^j)$. Assume the first case (that is depicted in Figure 10 for $j = 3$; the subsequence is F^3, F^4, F^5, F^6). Then we will consider the graph $M(n, \kappa_1, \kappa_2, 2)$ with $\kappa_1 = 2j + 1$ and $\kappa_2 = 2(n - 1 - j) + 1$ (in Figure 10 the point K is $(\kappa_1, \kappa_2) = (7, 1)$). This is always possible, that is, for every n, j we have $\kappa_1, \kappa_2 > 0$. But then we have for every k, $0 \le k \le n - 1$, that $F^{j+k} = (\kappa_1, \kappa_2) + (2(n - 1) + 2k, 4(n - 1) - 2k) = \vec{c}^{\,k}$, that is, the extracted subsequence $(F^j, F^{j+1}, \ldots, F^{j+n-1})$ is exactly the set of solution costs for $M(n, \kappa_1, \kappa_2, 2)$, $(\vec{c}^{\,0}, \vec{c}^{\,1}, \ldots, \vec{c}^{\,n-1})$. Since F^j is nd-selected over $(F^{j+1}, \ldots, F^{j+n-1})$, by Lemma 2 MOA* performs at least $n + \frac{n(n-1)}{2}$ node expansions on $M(n, \kappa_1, \kappa_2, 2)$.

Analogously we can prove the other case considering a graph of the form $M(n, \kappa_3, \kappa_4, 4)$. $\triangleleft$

From Lemmas 1 and 3 we obtain immediately the following result.

Theorem 1 *For every nd-selection rule depending only on $\vec{f}$ values, there exists a sequence of graphs $M_1, \ldots, M_n, \ldots$ such that:*

(i) Every M_n has $2n$ nodes and $3n - 2$ arcs.

(ii) MOA performs $\Theta(n)$ node expansions when applied to M_n and no heuristic information is given.*

(iii) MOA performs $\Omega(n^2)$ node expansions when applied to M_n and perfect heuristic information is given.*

6. Label Counts

The basic operation of MOA* is node expansion and every node expansion implies —in the general case— the joint expansion of several labels. However, other algorithms (e.g. NAMOA*, Mandow & Pérez de la Cruz, 2010a) use label expansion as the basic operation. For this reason it could be interesting to analyse MOA* also in terms of label expansions.

Lemma 4 *If the input of MOA* is a graph $M(n, \kappa_1, \kappa_2, \alpha)$ and $H(n) = \{\vec{0}\}$, then MOA* performs exactly $n^2 - n + 1$ label expansions.*

Proof. Consider again the reasoning for the proof of Lemma 1. It was then proved that, when selected for expansion, every even node $2i$ has i labels and every odd node $2i+1$ (with $i \geq 1$) has i labels. Adding all together we have $1 + 2\sum_{1\leq i\leq n-1} i = 1 + n(n-1) = n^2 - n + 1$ label expansions.
$\triangleleft$

Lemma 5 *For every $n \in \mathbb{N}$ there exists a graph $M(n, \kappa_1, \kappa_2, \alpha)$ such that when it is input to MOA* and $H(n) = H^*(n)$, then MOA* performs at least $n^2 + \frac{n(n+1)(n-1)}{4}$ label expansions.*

Proof. Every odd node $1, \ldots, 2n-1$ must have at termination n labels, and must be expanded at least once. That amounts to at least n^2 label expansions. On the other hand, consider even nodes $2, 4, \ldots, 2n-2$. They must also have n labels at termination. Consider again the reasoning for the proof of Lemma 2. It was then proved that: (i) the first time an even node $2i$ is expanded, it has $(n-i+1)$ labels; (ii) each time an even node $2i$ is expanded, it has exactly one more label. Therefore the number of label expansions for node $2i$ is $(n-i+1) + (n-i+2) + \ldots + (n-i+i) = \sum_{1\leq j\leq i}(n-i+j) = \frac{(n+1)i}{2}$. Adding togheter for all even nodes $2i$ with $1 \leq i \leq n-1$ we have $\sum_{1\leq i\leq n-1} \frac{(n+1)i}{2} = \frac{n(n+1)(n-1)}{4}$ label expansions. Adding now the expansions of odd and even nodes we have at least $n^2 + \frac{n(n+1)(n-1)}{4}$ q. e. d.
$\triangleleft$

From Lemmas 4 and 5 we obtain immediately the following result.

Theorem 2 *For every nd-selection rule depending only on $\vec{f}$ values, there exists a sequence of graphs $M_1, \ldots, M_n, \ldots$ such that:*

(i) Every M_n has $2n$ nodes and $3n-2$ arcs.

(ii) MOA performs $\Theta(n^2)$ label expansions when applied to M_n and no heuristic information is given.*

(iii) MOA performs $\Omega(n^3)$ label expansions when applied to M_n and perfect heuristic information is given.*

7. Conclusions and Future Work

This paper considers the performance of the MOA* multiobjective heuristic search algorithm. Results show that performance can degrade with better heuristic information. A class of problems is presented (multiobjective chain graphs) where the use of perfect heuristic information (a trivially consistent informed heuristic) does not result in a reduction in the number of node or label expansions performed by the algorithm. On the contrary, the performance of a perfectly informed version of the algorithm is worse than the performance of the blind version.

Multiobjective chain graphs formalize a not so infrequent situation in practical multiobjective search, when a sequence of nodes is traversed by at least two conflicting paths. Our analysis has revealed that when MOA* is combined with H^*, the best possible heuristic, the number of node expansions grows quadratically, while this number grows linearly when no heuristic information is used; and the number of label expansions grows cubically, while it grows quadratically when no heuristic information is provided.

This pathology, together with other results both theoretical (Mandow & Pérez de la Cruz, 2010b) and empirical (Machuca, Mandow, Pérez de la Cruz, & Ruiz-Sepúlveda, 2012), casts some doubts on the suitability of MOA* for performing heuristic multiobjective search. In general, other alternatives (such as NAMOA*) for which it has been proved (Mandow & Pérez de la Cruz, 2010a) that those pathological behaviours cannot arise, should be preferred.

Acknowledgments

Partially supported by Gobierno de España, grant TIN2009-14179. Partially funded by Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03018.

References

Boxnick, S., Klöpfer, S., Romaus, C., & Klöpper, B. (2010). Multiobjective search for the management of a hybrid energy storage system. In *IEEE International Conference on Industrial Informatics (INDIN)*, pp. 745–750.

Caramia, M., Giordani, S., & Iovanella, A. (2010). On the selection of k routes in multiobjective hazmat route planning. *IMA Journal of Management Mathematics*, *21*, 239–251.

Dasgupta, P., Chakrabarti, P., & DeSarkar, S. (1995). Utility of pathmax in partial order heuristic search. *Information Processing Letters*, *55*, 317–322.

Dasgupta, P., Chakrabarti, P., & DeSarkar, S. (1999). *Multiobjective Heuristic Search.* Vieweg, Braunschweig/Wiesbaden.

De Luca Cardillo, D., & Fortuna, T. (2000). Dea model for the efficiency evaluation of nondominated paths on a road network. *European Journal of Operational Research*, *121*(3), 549–558.

Delling, D., & Wagner, D. (2009). Pareto paths with SHARC. In *SEA*, pp. 125–136.

Dell'Olmo, P., Gentili, M., & Scozzari, A. (2005). On finding dissimilar Pareto-optimal paths. *European Journal of Operational Research*, *162*, 70–82.

Ehrgott, M. (2005). *Multicriteria Optimization.* Springer.

Fave, F. M. D., Canu, S., Iocchi, L., Nardi, D., & Ziparo, V. A. (2009). Multi-objective multi-robot surveillance. In *4th International Conference on Autonomous Robots and Agents (ICARA)*, pp. 68–73. IEEE.

Gabrel, V., & Vanderpooten, D. (2002). Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research*, *139*, 533–542.

Galand, L., & Perny, P. (2006). Search for compromise solutions in multiobjective state space graphs. In *Proc. of the XVII European Conference on Artificial Intelligence (ECAI'2006)*, pp. 93–97.

Galand, L., Perny, P., & Spanjaard, O. (2010). Choquet-based optimisation in multiobjective shortest path and spanning tree problems. *European Journal of Operational Research*, *204*(2), 303–315.

Galand, L., & Spanjaard, O. (2007). Owa-based search in state space graphs with multiple cost functions. In *FLAIRS Conference 2007*, pp. 86–91.

Hart, P., Nilsson, N., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Systems Science and Cybernetics SSC-4*, *2*, 100–107.

Klöpper, B., Ishikawa, F., & Honiden, S. (2010). Service composition with pareto-optimality of time-dependent QoS attributes. *Lecture Notes in Computer Science*, *LNCS 6470*, 635–640.

Lustrek, M., & Bulitko, V. (2008). Thinking too much: Pathology in path finding. In et al., M. G. (Ed.), *Proceedings of the European 18th Conference on Artificial Intelligence*, pp. 899–900. IOS Press.

Machuca, E., Mandow, L., Pérez de la Cruz, J. L., & Ruiz-Sepúlveda, A. (2010). An empirical comparison of some multiobjective graph search algorithms. In *KI'2010 - LNAI 6359*, pp. 238–245.

Machuca, E., Mandow, L., Pérez de la Cruz, J. L., & Ruiz-Sepúlveda, A. (2012). A comparison of heuristic best-first algorithms for bicriterion shortest path problems. *European Journal of Operational Research*, *217*, 44–53.

Machuca, E., & Mandow, L. (2011). Multiobjective route planning with precalculated heuristics. In *Proc. of the 15th Portuguese Conference on Artificial Intelligence (EPIA 2011)*, pp. 98–107.

Mandow, L., & Pérez de la Cruz, J. L. (2003). Multicriteria heuristic search. *European Journal of Operational Research*, *150*, 253–280.

Mandow, L., & Pérez de la Cruz, J. L. (2005). A new approach to multiobjective A* search. In *Proc. of the XIX Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pp. 218–223.

Mandow, L., & Pérez de la Cruz, J. L. (2010a). Multiobjective A* search with consistent heuristics. *Journal of the ACM*, *57*(5), 27:1–25.

Mandow, L., & Pérez de la Cruz, J. L. (2010b). A note on the complexity of some multiobjective A* search algorithms. In *ECAI 2010*, pp. 727–731.

Martelli, A. (1977). On the complexity of admissible search algorithms. *Artificial Intelligence*, *8*, 1–13.

Mouratidis, K., Lin, Y., & Yiu, M. (2010). Preference queries in large multi-cost transportation networks. In *Proceedings - International Conference on Data Engineering*, pp. 533–544.

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

Nau, D. S. (1982). An investigation of the causes of pathology in games. *Artificial Intelligence*, *19*(3), 257–278.

Nau, D. S., Lustrek, M., Parker, A., Bratko, I., & Gams, M. (2010). When is it better not to look ahead?. *Artificial Intelligence*, *174*(16–17), 1323–1338.

Pearl, J. (1984). *Heuristics*. Addison-Wesley, Reading, Massachusetts.

Perny, P., & Spanjaard, O. (2002). On preference-based search in state space graphs. In *Proc. Eighteenth Nat. Conf. on AI*, pp. 751–756. AAAI Press.

Perny, P., & Spanjaard, O. (2005). A preference-based approach to spanning trees and shortest paths problems. *European Journal of Operational Research*, *162*, 584–601.

Refanidis, I., & Vlahavas, I. (2003). Multiobjective heuristic state-space search. *Artificial Intelligence*, *145*, 1–32.

Stewart, B. S., & White, C. C. (1991). Multiobjective A*. *Journal of the ACM*, *38*(4), 775–814.

Wu, P.-Y., Campbell, D., & Merz, T. (2009). On-board multi-objective mission planning for unmanned aerial vehicles. In *IEEE Aerospace Conference Proceedings*, pp. 1–10.

Wu, P.-Y., Campbell, D., & Merz, T. (2011). Multi-objective four-dimensional vehicle motion planning in large dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *41*(3), 621–634.

Ziebart, B., Dey, A., & Bagnell, J. (2008). Fast planning for dynamic preferences. In *ICAPS 2008 - Proceedings of the 18th International Conference on Automated Planning and Scheduling*, pp. 412–419.