Multiobjective route planning with precalculated heuristics *

E. Machuca and L. Mandow

Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga
29071 Málaga (Spain)
{machuca, lawrence}@lcc.uma.es

Abstract. This paper describes the application of multiobjective heuristic search algorithms to the problem of route planning in road maps. Multiobjective search reveals the trade-off between objectives and allows for more informed decisions than single objective search but, at the same time, it is known to be more complex computationally. The efficiency of existing search algorithms is known to depend on factors such as correlation between objectives and label selection schemes. The paper shows the application of multiobjective search techniques to route planning problems involving two relatively uncorrelated objectives: economic cost (including fuel and tolls) and travel time. The impact of heuristic information and label selection in search efficiency are analyzed.

1 Introduction

Route planning in road maps is a current research topic among transportation problems. Most recent contributions concentrate on single objective problem formulations [15] [2]. Current techniques generally rely on exhaustive precalculations that can be later used to speed up route queries. Travel time, distance or economic cost are frequent objectives to be minimized, and most current road planners offer the opportunity to optimize each of them individually.

Multiobjective analysis is concerned with the simultaneous optimization of different frequently conflicting and noncommesurate objectives [5]. Rather than a single optimal solution, multiobjective analysis provide a set of efficient or Pareto optimal solutions. These represent the optimal trade-offs between the objectives under consideration. This information generally allows more informed and realistic decisions. On the other side, multiobjective algorithms are known to be more complex computationally than single objective ones. Therefore, the development of efficient multiobjective search techniques is an active area of research.

Some attempts have been carried out to extend current exact route planning techniques to multiobjective problems [5]. However, these usually rely on extensive uninformed multiobjective preprocessing, which is computationally impractical in large maps. This approach has been successfully applied to maps with 77,000 nodes.

* This work is funded by/Este trabajo está financiado por: Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03018
This paper evaluates the application of current multiobjective heuristic search techniques to a moderately large road map (264,346 nodes). In particular, two exact heuristic search algorithms are evaluated, NAMOA* [12] and Tung & Chew’s [18]. A precalculated multiobjective heuristic proposed by Tung and Chew is used in both algorithms, and its performance compared to blind search. Two objectives are optimized simultaneously: travel time, and economic cost, which includes fuel cost and highway tolls. The efficiency of multiobjective search algorithms is known to depend on the particular label selection strategy. Several such strategies are evaluated over a set of randomly selected problems: lexicographic, linear, and heuristic.

The analysis reveals that heuristic NAMOA* with a linear selection rule is the most efficient alternative. The exact solution to all tested problems could be found with available resources. However, current time requirements are only reasonable for off-line route planning on large sized maps.

The paper is organized as follows. Section 2 summarizes related work and previous results in single and multiobjective search. Section 3 describes the evaluation performed and section 4 discusses the experimental results. Finally, some conclusions and future work are outlined.

2 Previous work

Route planning in road maps can be in general modelled as a shortest path problem, where arcs represent roads, and nodes represent road junctions.

2.1 Single-objective search

Probably the best known and most widely used algorithm to solve the single-objective shortest path problem is Dijkstra’s best-first algorithm [3]. However, heuristic estimates can be applied to accelerate the search for an optimal path to a destination node. The $A^*$ algorithm [9] incorporates heuristic information in a characteristic evaluation function $f(n) = g(n) + h(n)$ used to select the next best promising node $n$. The cost $g(n)$ of the currently best path known to $n$ is combined with $h(n)$, an heuristic estimate of the cost of a path reaching the goal from $n$. When $h(n)$ is optimistic (i.e. a lower bound of the optimal cost), $A^*$ is guaranteed to find optimal solutions (admissibility). Heuristics can be used to speed up search, since, under reasonable assumptions, the algorithm’s efficiency is known to improve with the quality of heuristic information [15].

Pearl [16] analyzed the relative advantage of $A^*$ (using an Euclidean distance heuristic) over Dijkstra’s algorithm in road maps. Some other empirical studies were performed with real road networks using $A^*$ in the past [12]. New techniques have been developed in the recent years to further improve search efficiency, like precomputing distance bounds [2] or the use of bidirectional $A^*$ [13].

However, many recent works concentrate on speed up techniques for Dijkstra’s algorithm. In general, these techniques precompute some information to speed up later shortest-path queries. The general idea is to achieve a fast query with practical time and memory preprocessing. Good overviews of these techniques can be found elsewhere [16] [2].
2.2 Multiobjective heuristic search

Real-world decision problems frequently need the consideration of several criteria at the same time. For example, in the context of hazardous material transportation, the best route may involve the consideration of a short route that, at the same time, minimizes the exposure of population to risk \[6\].

In multiobjective search, arcs are labelled with vector costs, where each element in the vector stands for a different relevant attribute. For example, in the biobjective case each arc from \(n\) to \(m\) is labelled with \(c(n, m) = (c_1, c_2)\). The use of cost vectors in multiobjective problems induces a partial order relation called dominance (\(\prec\)),

\[
\forall v, v' \in \mathbb{R}^q, \quad v \prec v' \iff \forall i(1 \leq i \leq q), \quad v_i \leq v'_i \land v \neq v' \quad (1)
\]

where \(v_i\) denotes the \(i\)-th component of vector \(v\). Many different nondominated paths may reach every node in multiobjective graph search problems, defining the so-called Pareto front (see figure \(\text{I}\)). Therefore, multiobjective algorithms try to find the set of all nondominated (Pareto-optimal or efficient) solution paths. This makes the problem much more complex than single-objective search.

Fig. 1: A typical Pareto-front in cost space. Alternatives A, B and C do not dominate each other. Each of the other alternatives is dominated by at least one of A, B, or C.

Extensions of Dijkstra’s algorithm to the multiobjective case are due to Hansen [8] (biobjective case) and Martins [13] (general case). Regarding heuristic search, several multiobjective extensions of A* have been proposed. These are MOA* (Multi-objective A*) [12], NAMOA* (New Approach to Multi-objective A*) [12] and Tung & Chew’s algorithm [18]. In particular, NAMOA* has been proven optimal among admissible algorithms and to strictly dominate MOA*. Therefore, this latter algorithm is not considered in this paper.

In these algorithms, each node \(n\) may be reached by a number of distinct nondominated paths. Nodes are labelled with their costs. Two sets of labels are used for each node \(n\): \(G_{op}(n)\), that stands for unexplored or open labels, and \(G_{cl}(n)\), that stands for
explored or closed labels. These algorithms also accept a heuristic vector estimate \( h(n) \) of the cost of paths from \( n \) to the goal. A heuristic evaluation \( f \) can be calculated for each open label \( g \in G_{op}(n) \), simply adding the label with the heuristic estimate \( h(n) \), analogously to A*.

In order to guarantee that all nondominated solutions are found, multiobjective algorithms need to select at each iteration an open label with a nondominated heuristic evaluation. Since there may be many such nondominated open labels, some tie-breaking procedure must be introduced. For example, the lexicographic optimum of a set of vectors is known to be a nondominated element in the set (lexicographic selection rule). Also, if all vector components are added, the element with the smallest value is also known to be nondominated (linear selection rule). NAMOA* accepts any of such schemes. However, Tung & Chew’s algorithm uses a special linear selection rule that adds a particular precalculated heuristic estimate to the components of each label evaluation vector (see [18] for details).

A recent analysis on uninformed multiobjective search has suggested that a linear aggregation rule is more effective than a lexicographic one [10]. To the author’s knowledge, no analogous study has been reported on heuristic multiobjective algorithms.

Regarding the precalculated heuristic vector \( h(n) = (h_1, h_2, \ldots, h_q) \), Tung & Chew proposed that the individual \( h_i \) values be precalculated through search. In particular, the graph is reversed and optimal costs from the goal node to all other nodes in the graph are precomputed using Dijkstra’s algorithm, once for each objective under consideration. This approach is practical, since the single-objective searches can be generally calculated much more efficiently than the overall multiobjective search.

Several authors have addressed the problem of multiobjective search in road maps with different techniques. For example, Caramia et al. [11] applied Martin’s algorithm to hazardous material transportation in a network with over 600 nodes. The work of Delling and Wagner [3] considered extensive precalculations with Martin’s algorithm on a network with over 77,000 nodes, but resorted to an approximation scheme for larger networks.

The next section describes the application of NAMOA* and Tung & Chew’s algorithm to a network with over 200,000 nodes.

### 3 Experiments

This section describes the experimental setting devised to test the application of multiobjective heuristic search to route planning. A set of problems were defined over a publicly available road map of the New York City area used for the 9th DIMACS Implementation Challenge on Shortest Paths. The road map defines a graph where arcs represent roads, and nodes represent road junctions. These were originally taken from the 2000 U.S. Census Bureau’s TIGER/Line® files (Topologically Integrated Geographic Encoding and Referencing system). The map includes 264,346 nodes and 730,100 arcs.

Two different arc costs are publicly available for this map: physical distance (in decimeters), and travel time. The latter cost was originally obtained from the application

of time factors to arc distances according to the TIGER/Line road classification scheme: A (primary highway), B (primary road), C (secondary and connecting road), and D (local, neighborhood, and rural road). Time factors are 1.0, 0.8, 0.6, and 0.4 respectively.

The experiments reported below consider the simultaneous minimization of travel time and travel cost. The original time values of the DIMACS Challenge were used in the experiments. The travel cost attribute was calculated from available information according to the following considerations,

- Travel cost results from the addition of fuel cost and tolls.
- All roads type A are set to pay a toll of 1.86 cents per Kilometer (3 cents per mile). All other roads are toll-free. A rendering of the roads type A of the map is shown in figure 2.
- Fuel cost depends on fuel price, fuel efficiency and travelled distance.
- Road type. Fuel efficiency depends on road type, and particularly on its allowed maximum speed. We associate current speed limits in the NY City area to every TIGER/Line road type as shown in table 1. Road types were calculated using the physical distance, travel time, and time factors information.
- Fuel efficiency. Fuel efficiency is usually measured in miles per gallon (mpg) or litres per 100 kilometer (l/100Km). Fuel efficiency depends heavily on travel speed and vehicle type (and also on the particular model and production year). Instead of conducting our experiments for a particular car model, we used general gas mileage values provided by a standard chart from the U.S. Department of Energy 
  2. These values are shown in table 2. This chart reflects the fact that, in general, maximum fuel efficiency is usually achieved at speeds in the range 80-90 Km/h (50-55 mph).
- Fuel price. The price of fuel was set to a current value in the New York City area of 375 cents/gallon (99 cents/litre).
- In summary, fuel cost in cents was calculated for each arc in the graph according to the fuel spent travelling its distance with its assigned general fuel efficiency rate.

While these calculated costs may not reflect precisely actual travel costs, they are sufficiently realistic for experimentation purposes. The computation of the Pearson's coefficient over the two objectives gives a value of 0.16, i.e. the objectives are not linearly correlated. This is therefore a difficult multiobjective problem. Figures 2 and 3 display the set of toll highways, and all roads in the map respectively.

A set of 20 route planning problems were generated selecting random origin and destination nodes using an uniform distribution.

Regarding the algorithms, NAMOA* was evaluated with both a lexicographic and a linear selection rule. Additionally, the algorithm was run without heuristic information (i.e. \( \forall n h(n) = 0 \)), and using the precalculated Tung-Chew heuristic as described in section 2. Tung & Chew’s algorithm was run only with heuristic information, and using its special linear selection rule, as required in the original description of the algorithm.

---

2 http://www.fueleconomy.gov/feg/drivehabits.shtml
3 A file in DIMACS format with the calculated travel costs is available from the authors
Fig. 2: Rendering of primary (toll) highways of the New York City map.

Fig. 3: Rendering of the New York City map.
### Table 1: Speed limits and fuel efficiency values associated to each road type.

<table>
<thead>
<tr>
<th>Road type</th>
<th>Limit (mph)</th>
<th>Limit (Km/h)</th>
<th>mpg</th>
<th>l/100Km</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>65</td>
<td>105</td>
<td>27</td>
<td>8.71</td>
</tr>
<tr>
<td>B</td>
<td>55</td>
<td>89</td>
<td>30</td>
<td>7.84</td>
</tr>
<tr>
<td>C</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>7.84</td>
</tr>
<tr>
<td>D</td>
<td>30</td>
<td>48</td>
<td>29</td>
<td>8.11</td>
</tr>
</tbody>
</table>

4 Results

Figure 4 shows the time requirements for the five algorithmic instances under consideration, i.e. blind and heuristic NAMOA* with lexicographic selection (NAMOA-LEX), blind and heuristic NAMOA* with linear selection (NAMOA-LIN), and Tung & Chew’s algorithm. The ordinate axis $x$ shows problem indexes for each instance in the 20 problem set. Instances are ordered according to increasing time taken by NAMOA-LIN with heuristic. The value for abscissa $y$ (time in seconds) is shown in a logarithmic scale.

For practical reasons, instances were solved with a twelve-hour time limit. In the case that an algorithm could not solve an instance, values are not displayed in the figure for that particular heuristic-algorithm combination. In addition, values under 1 second obtained by the heuristic algorithms TC, NAMOA-LIN and NAMOA-LEX in problem instances NY 15 and NY 3 are not displayed for the sake of clarity in figure 4.

Heuristic NAMOA* with a linear selection rule is clearly the best option. It is worth noting that the special selection heuristic used by Tung and Chew’s algorithm does not seem to provide any practical advantage. This confirms previous analysis which suggest that using additional heuristic information in label selection can in fact degrade time performance [11].

Table 2a shows the number of problem instances that were unsolvable within the time limit for each algorithm. A set of 4 problem instances could not be solved within the time limit regardless of the algorithm. These problem instances were solved without time limit with heuristic NAMOA-LIN, which is identified in the analyses as the most effective alternative. All of them could be solved with the available resources. Table 2b shows the time taken to solve each of these instances. The hardest one, NY 4, was solved in about 127 hours.

Table 3 shows in the first two columns the average speedup of NAMOA* with respect to TC algorithm and blind search, for each one of the two strategies used for selection, LEX and LIN. The results presented in figure 4 indicate that the linear strategy is the most suitable one for all cases, with a speedup of 1.55 over Tung & Chew’s algorithm, and of 693 over blind NAMOA with linear selection. Additionally, with heuristic NAMOA*, the speedup of linear versus lexicographic selection was 1.66. With uninformed NAMOA*, the speedup of linear versus lexicographic was 1.27. Only the problems that could be solved within the runtime limit were taken into account for the average speedups.

---

4 Notice that uninformed NAMOA* with lexicographic selection is equivalent to Martins’ algorithm
<table>
<thead>
<tr>
<th>Problem Id</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blind NAMOA-LIN</td>
<td>15 3 7 11 1 10 18 9 6 16 17 5 12 13 2 14 20 19 8 4</td>
</tr>
</tbody>
</table>

Fig. 4: Time requirements
5 Conclusions and future work

The paper presents an analysis of heuristic search algorithms for multiobjective route planning problems. The analysis involves the consideration of two linearly uncorrelated objectives (travel time and travel cost) defined over a relatively large road map, with 264,346 nodes and 730,100 arcs.

The analysis reveals that uninformed search algorithms are in general not practical for this kind of problems. This is a serious objection for approaches that rely on extensive uninformed precalculations.

Regarding heuristic search approaches, two algorithms were analyzed, Tung & Chew’s and NAMOA*. The latter was tested with lexicographic and linear selection rules. The analysis confirms that the linear selection rule is more efficient than the lexicographic one. However, the special heuristic linear selection used by Tung & Chew’s algorithm was found to perform worse. In fact, this strategy is even worse than heuristic lexicographic NAMOA* for the hardest problems.

While heuristic NAMOA* with linear selection was able to solve all problems with the available resources, this kind of search is limited to off-line route planning applications.

Future work involves the evaluation of these algorithms on related applications, like hazmat transportation, and the development of more efficient heuristics.

References