

Fuel-cache site-selection for polar research: A Summary of Results

Mark Dietz
Department of Computer Science
University of Minnesota
mdietz@cs.umn.edu

Shashi Shekhar
Department of Computer Science
University of Minnesota
shekhar@cs.umn.edu

ABSTRACT

Scientists conducting polar research in Antarctica must contend with harsh environmental conditions that constrain their movements and raise their costs. One on going challenge is choosing cache sites for aircraft refueling. Given a data-gathering mission (e.g. set of flight destinations), aircraft fuel-consumption model, and infrastructure (e.g. base and cache-sites), the Fuel-Cache Site-Selection (FCSS) problem identifies the optimal use of cache sites to fulfill the mission. The FCSS problem is important for planning expeditions in infrastructure-poor areas for scientific or military purposes. However, the FCSS problem is computationally challenging due to interaction across different flight-routes. Related approaches from literature concerning routing are inadequate due to assumptions about the cost of providing infrastructure. This paper proposes heuristics and a filter-and-refine based exact algorithm, evaluation using analytical and experimental methods, and a case study with end-users, e.g. polar scientists.

Categories and Subject Descriptors

I.2.1 [Applications and Expert Systems]: Route Finding

General Terms

Algorithms

Keywords

Routing, Assignment, Optimization.

1. INTRODUCTION

Scientists conducting polar research in Antarctica must contend with harsh environmental conditions that constrain their movements and raise their costs. One on going challenge is choosing cache sites for aircraft refueling. Given a data-gathering mission (e.g. set of flight destinations), aircraft fuel-consumption model, and infrastructure (e.g. bases and cache-sites), the Fuel-Cache Site-Selection (FCSS) problem identifies the optimal use of infrastructure to fulfill the mission. Each destination must be visited by a separate flight due to the amount of time that must be spent at each destination and constraints on the total time spent in the field. Thus a solution to the FCSS problem specifies a path from the home point to the flight destination and back to the home point. Some points can be reached directly. Others require a refueling stop, which can only be made at predefined cache-sites. All fuel taken from the cache-sites must first be placed there by a separate refueling flight. Therefore the FCSS problem consists of minimizing the fuel consumption of two types of flights: research flights to the specified destinations of interest and refueling flights to place fuel at the cache sites in support of the research flights.

The FCSS problem is important for planning expeditions in infrastructure-poor areas for scientific or military purposes. The

logistics costs for supporting these missions are very high due to the cost incurred establishing the necessary infrastructure to support the mission. There is much value in reducing these costs.

However, the FCSS problem is computationally challenging due to interaction across different flight-routes. The choice of refueling site for a particular research point may vary based on whether there is left over fuel from a previous refueling flight at any of the cache sites.

Related Work: Prior research on problems similar to FCSS falls into two categories: vehicle routing with fuel constraints and multi-depot vehicle routing.

Research into vehicle routing with fuel constraints ([1] and [2]) attempts to find the least cost path between two points in a graph for a vehicle with a limited fuel tank capacity. The vehicle must stop to refuel at designated refueling stations. The graph edges are weighted by the fuel required and each refueling station is a node on the graph with a fixed fuel price.

Multi-depot vehicle routing ([3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] and [15]) is the problem of transporting materials to customers using a number of depots. Customers are assigned a depot and vehicles are routed to satisfy the customers' demands while attempting to minimize the costs of fuel, vehicles and drivers.

In the FCSS problem, the fuel cache sites take the role of the refueling stations or depots and the research points take the role of the destination points or customers. However, the approaches in the literature assume the cost of choosing a cache site for a research point is independent of the other research points using that site. In FCSS, the cost of fuel at the fuel cache site is the cost to transport the fuel there. This cost varies based on which research points are assigned to that fuel cache.

Our Contributions: This paper defines the fuel cache site selection problem (FCSS). We believe the problem is NP-hard due to similarities with other routing problems and propose two heuristics. The paper also provides an algorithm to find the optimal solution using properties of the problem to filter the possible cache site choices for each research point and performing an exhaustive search of the remaining choices. Finally, it presents a case study showing how these solution methods combined with a visualization tool (Google Earth) can be used by human planners to assess the fuel cost impacts of adding a new cache site or removing an existing site.

Outline: The rest of the paper is organized as follows. Section 2 formally defines FCSS and provides an example. Section 3 presents our methods for solving the selection problem. Section 4 shows an analytical and experimental comparison between the solution methods on real and synthetic data. Section 5 discusses our conclusions.

2. PROBLEM DEFINITION

2.1 Definitions

Definition 2.1.1: Aircraft model – We assume the aircraft used for all flights can be specified by the following model:

- Maximum Fuel Capacity
- Fuel Burned per Mile
- Minimum Fuel Reserve
- Base Operational Weight: weight of the aircraft and crew without cargo, passengers, or fuel
- Maximum Operational Weight: weight of the aircraft plus crew, cargo, passengers, and fuel

Definition 2.1.2: $fuelCost(h, p)$: fuel cost for a round trip from home point h to point p – either a cache site or a research point.

Definition 2.1.3: $fuelCost(h, r_i, c_j)$: fuel cost for a round trip from home point h to research point r_i using fuel cache c_j to refuel.

Definition 2.1.4: $refuelCost(h, r_i, c_j)$: fuel cost to place enough fuel at cache c_j to support a round trip research flight to r_i that uses c_j to refuel. For example, if the research flight to r_i needs to take 1000 lbs of fuel from the site c_j and the maximum payload for a refueling flight to c_j is 800 lbs, then $refuelCost(h, r_i, c_j)$ is twice $fuelCost(h, c_j)$.

2.2 Formal Problem Definition

Given:

- A single home point h represented by latitude / longitude coordinates.
- Set $C = \{c_1, c_2, \dots, c_m\}$ of fuel cache sites represented as latitude / longitude coordinates.
- Set $R = \{r_1, r_2, \dots, r_n\}$ research points of interest represented as latitude / longitude coordinates.
- Aircraft model.

Find:

- For every $r_i \in R$ find a feasible path from h to r_i and back to h . The feasible path contains no other research points and at most one stop to refuel at a site $c_{i1} \in C$ en route from h to r_i and at most one stop to refuel at $c_{i2} \in C$ en route from r_i to h .
- The amount of fuel to place at each cache site.
- The number of refueling flights needed to place fuel at each cache site.

Objective:

- Minimize the total fuel consumed by the research flights (flights to the points in R) and the refueling flights (flights to the cache sites in C to deposit the fuel used to refuel during the research flights).

Constraints:

- All refueling is performed at h or a point in C .
- h is assumed to be an infinite fuel source.
- All fuel taken from a site $c_i \in C$ must be flown from h to c_i using a refueling flight.
- A flight is either a refueling flight for placing fuel or research flight for visiting a research point, not both.
- Refueling flights carry fuel in drums of a specified size and weight. For example, each drum weights 400 lbs and contains 350 lbs of fuel.
- Sites in C have the capacity to store an infinite number of fuel drums.

2.3 Example

Figure 1 shows an example problem instance with $R = \{r_1, r_2, r_3, r_4\}$ and $C = \{c_1, c_2, c_3\}$. In this example, r_1 is close enough to h to be reached directly without stopping to refuel. r_2 can only be reached by refueling at c_1 on the outbound flight and refueling at c_1 again on the home bound flight. Therefore these assignments must be included in any optimal solution. The only choices to be made for this problem instance are selecting cache sites for r_3 and r_4 . Point r_3 can be visited with just one stop to refuel at either c_2 (closest) or c_3 (a little farther), but c_1 is too far away. Point r_4 can be visited with just one refuel stop at c_3 (closest) or c_2 (a little farther). It can also be reached by refueling at c_1 , but at a much larger cost. Table 1 shows the costs for r_3 and r_4 for assignments to c_2 and c_3 .

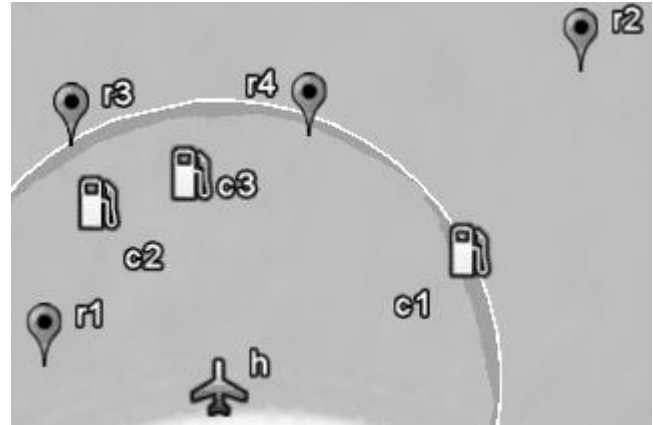


Figure 1: Example of cache selection problem with four research points and three fuel caches. Circle indicates maximum round trip flight range without refueling.

Table 1: Feasible solutions for r_3 and r_4 in example problem

Solution #	Research Point	Cache site to refuel	Fuel consumed by research flights	Total Plan for r_3, r_4 Cost With Fuel Placement
1	r_3	c_2	2807	9090
1	r_4	c_3	2774	
2	r_3	c_2	2807	9590
2	r_4	c_2	3245	
3	r_3	c_3	2850	7364
3	r_4	c_3	2774	
4	r_3	c_3	2850	9534
4	r_4	c_2	3245	

Table 1 shows that Solution #1 optimizes the fuel consumed by the flights to the research points and would be the best choice if there were no cost for fuel placement. However, Solution #3 is the optimal solution when fuel placement costs are considered. In Solution #3, fuel placement costs are lower because fuel needs to be placed at only one location to support r_3 and r_4 thus allowing consolidation of refueling flights. Note Solution #2 also chooses the same site for r_3 and r_4 , but these flights require more fuel to be placed at c_2 than can be carried in one refueling flight so two refueling flights are necessary.

2.4 Contrast FCSS Routing with other Routing

In many routing problems, such as the Traveling Salesman Problem, a route must be found that can visit many points of interest. By contrast in FCSS we must find a separate route from the home point to each point of interest and back to the home

point. The route cannot include any other research points. This constraint arises from real world conditions in polar research. Scientists visiting a research point must spend several hours taking measurements and installing sensors. There is no time to visit a second research point in a day and Antarctica's inhospitable climate makes an overnight stay in the field impossible. Other problems that would include this constraint are:

- an installation task where the vehicle can carry the materials for only one installation at a time
- a research task where the tools needed are very specific to the particular task and so not all tools can be carried at once
- a transportation task where vehicle capacity is limited to the items for one customer

3. CACHE SELECTION METHODS

Due to the belief that this problem is NP-hard, there is not likely a scalable algorithm to find an exact solution. We present two heuristics and one exact algorithm for solving the FCSS problem as presented in section 2.2. The first step of each is to identify any research points that are close enough to the home point h to not need a refueling stop at a cache site. These points are assigned a direct flight and then ignored. Even though we have shown that finding the globally optimal solution requires the cache selections to be interdependent, the heuristics each make different assumptions that allow the selections to be treated independently. The exact algorithm identifies bounds on the solution and filters out choices that do not fit within these bounds. It then performs an exhaustive search of the remaining choices. Once assignments are made, all three methods tally the total fuel required at each cache site and assign the minimum number of refueling flights needed to place the fuel at each cache site. This allows some refueling flights to be consolidated. However, only the exhaustive search considers the gains from consolidating refueling flights when making the cache site selections.

3.1 Heuristic: Research Flights Only

Figure 3 shows pseudo-code for the Research Flights Only heuristic. This heuristic is the most similar to the current manual process for solving the FCSS problem. For each research point $r_i \in R$, the heuristic chooses the cache $c_j \in C$ that minimizes $fuelCost(r_i, c_j)$. Ignoring the cost of fuel placement will provide an under estimate of the total fuel required to visit a research point. When executed with the input from example 2.3 this heuristic gives Solution #1 from Table 1. An execution trace follows:

- Only r_1 can be reached directly from h . Assign a direct flight for r_1 .
- r_2 can only be reached by refueling at c_1 . Assign c_1 to r_2 .
- r_3 can be reached by refueling at c_2 with $fuelCost(h, r_3, c_2) = 2807$ or c_3 with $fuelCost(h, r_3, c_3) = 2850$. Assign c_2 to r_3 .
- r_4 can be reached by refueling at c_1 with $fuelCost(h, r_4, c_1) = 3462$ or c_2 with $fuelCost(h, r_4, c_2) =$

3132 or c_3 with $fuelCost(h, r_4, c_3) = 2774$. Assign c_3 to r_4 .

- The amount of fuel needed at each site is: $c_1 = 2213$ lbs., $c_2 = 357$ lbs., and $c_3 = 394$ lbs.
- Refueling flights to c_1 can deposit a maximum of 700 lbs. of fuel per flight so c_1 requires 4 refueling flights.
- Refueling flights to c_2 can deposit a maximum of 1050 lbs. of fuel per flight so c_2 requires 1 refueling flight.
- Refueling flights to c_1 can deposit a maximum of 1400 lbs. of fuel per flight so c_1 requires 1 refueling flight.
- The total cost is $fuelCost(h, r_1) + fuelCost(h, r_2, c_1) + 4 \times fuelCost(h, c_1) + fuelCost(h, r_3, c_2) + fuelCost(h, c_2) + fuelCost(h, r_4, c_3) + fuelCost(h, c_3) = 24681$

```

ResearchFlightsOnly {
  foreach  $r_i$  in  $\{r_1, \dots, r_n\}$ 
    minFuelCost =  $\infty$ ;
    foreach  $c_j$  in  $\{c_1, \dots, c_m\}$ 
      if ( $fuelCost(h, r_i, c_j) < minFuelCost$ )
        minFuelCost =  $fuelCost(h, r_i, c_j)$ ;
        cacheSiteAssignment[ $r_i$ ] =  $c_j$ ;
    return cacheSiteAssignment[];
}

```

Figure 3: Pseudo-code for Research Flights Only

3.2 Heuristic: Independent Refueling Flights

Figure 4 shows pseudo-code for the Independent Refueling Flights heuristic. For each research point $r_i \in R$, this heuristic chooses the cache $c_j \in C$ that minimizes $fuelCost(h, r_i, c_j) + refuelCost(h, r_i, c_j)$. The cache selection does not take into account any opportunity for combining flights to place fuel at cache sites. This is not optimal because if the fuel needed for research flights to two research points that use the same cache can be supported with only one refueling flight, the cost will be counted against both research flights when evaluating alternatives. When executed with the input from example 2.3 this heuristic gives Solution #1 from Table 1. Note that while in this case Independent Refueling Flights chooses the optimal solution, this is not the case in general. An execution trace follows:

- Only r_1 can be reached directly from h . Assign a direct flight for r_1 .
- r_2 can only be reached by refueling at c_1 . Assign c_1 to r_2 .
- r_3 can be reached by refueling at c_2 with $fuelCost(h, r_3, c_2) + refuelCost(h, r_3, c_2) = 4575$ or c_3 with $fuelCost(h, r_3, c_3) + refuelCost(h, r_3, c_3) = 4520$. Assign c_3 to r_3 .
- r_4 can be reached by refueling at c_1 with $fuelCost(h, r_4, c_1) + refuelCost(h, r_4, c_1) = 8090$ or c_2 with $fuelCost(h, r_4, c_2) + refuelCost(h, r_4, c_2) = 5014$ or c_3 with $fuelCost(h, r_4, c_3) + refuelCost(h, r_4, c_3) = 4514$. Assign c_3 to r_4 .
- The amount of fuel needed at each site is: $c_1 = 2213$ lbs., $c_2 = 0$ lbs., and $c_3 = 794$ lbs.
- Refueling flights to c_1 can deposit a maximum of 700 lbs. of fuel per flight so c_1 requires 4 refueling flights.
- c_2 supplies no fuel so needs no refueling flights.
- Refueling flights to c_1 can deposit a maximum of 1400 lbs. of fuel per flight so c_1 requires 1 refueling flight.
- The total cost is $fuelCost(h, r_1) + fuelCost(h, r_2, c_1) + 4 \times fuelCost(h, c_1) + fuelCost(h, r_3, c_3) + fuelCost(h, c_3) + fuelCost(h, r_4, c_3) + fuelCost(h, c_3) = 22955$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

```

IndependentRefuelingFlights {
  foreach  $r_i$  in  $\{r_1, \dots, r_n\}$  {
    minFuelCostIncludingRefueling =  $\infty$ ;
    foreach  $c_j$  in  $\{c_1, \dots, c_m\}$  {
      if (fuelCost( $h, r_i, c_j$ )
          + refuelingCost( $h, r_i, c_j$ )
          < minFuelCostIncludingRefueling)
        minFuelCostIncludingRefueling =
          fuelCost( $h, r_i, c_j$ )
          + refuelingCost( $h, r_i, c_j$ );
      cacheSiteAssignment[ $r_i$ ] =  $c_j$ ;
    }
  }
  return cacheSiteAssignment[];
}

```

Figure 4: Pseudo-code for Independent Refueling Flights

3.3 Algorithm: Exhaustive Search With Filter

Figure 5 shows pseudo-code for the Exhaustive Search With Filter algorithm. This algorithm finds the upper bound for the total fuel consumption contribution for each research point and a lower bound for the fuel consumption contribution for the assignment of each cache site to each research point. These are shown in Lemma 3.3.1 and 3.3.2

Lemma 3.3.1: The upper bound on the fuel cost contribution for r_i is the amount of fuel required to visit r_i without considering other research points:

$$\text{MINIMUM}\{\forall c_j \in \text{caches} | \text{fuelCost}(h, r_i, c_j) + \text{refuelCost}(h, r_i, c_j)\}$$

Proof: Suppose there exists an optimal solution S where the contribution from r_i is greater than the proposed upper bound. Construct S' by replacing the flight to r_i and supporting refueling flights in S with a flight to r_i using c_j to refuel and supporting refueling flights. Then the total cost of S' is less than the total cost of S , a contradiction. Thus the contribution from r_i in the optimal solution S is less than the proposed upper bound.

Lemma 3.3.2: Lower bound on fuel cost contribution for r_i using cache site c_j is $\text{fuelCost}(h, r_i, c_j) + \text{refuelCost}(h, r_i, c_j) - \text{fuelCost}(h, c_j)$. This is the lower bound because there is at most one refueling trip's payload of fuel left over at c_j from refueling trips to support other research points. Thus refueling at the same site as other research flights saves at most one refueling flight.

Using the upper bound of $\text{fuelCost}(h, r_i, c_j) + \text{refuelCost}(h, r_i, c_j)$ we form the list of possible assignments for each $r_i \in R$. A cache c_j' is removed from the list of possible assignments if its lower bound $\text{fuelCost}(h, r_i, c_j') + \text{refuelCost}(h, r_i, c_j') - \text{fuelCost}(h, c_j')$ is higher than the upper bound $\text{fuelCost}(h, r_i, c_j) + \text{refuelCost}(h, r_i, c_j)$. Then all combinations of the choices are examined to find the optimal solution. When executed with the input from example 2.3, this algorithm gives Solution #1 from Table 1. An execution trace follows:

- Only r_1 can be reached directly from h . Assign a direct flight for r_1 .
- r_2 can only be reached by refueling at c_1 . Assign c_1 to r_2 .
- The maximum contribution from r_3 (Lemma 3.3.1) is $\text{fuelCost}(h, r_3, c_3) + \text{refuelCost}(h, r_3, c_3) = 4520$.
- The minimum contribution from r_3 with c_2 (Lemma 3.3.2) is $\text{fuelCost}(h, r_3, c_2) + \text{refuelCost}(h, r_3, c_2) - \text{fuelCost}(h, c_2) = 2807 < 4520$. c_2 cannot be filtered, so optimal selection for r_3 is in $\{c_2, c_3\}$.

- The maximum contribution from r_4 is $\text{fuelCost}(h, r_4, c_3) + \text{refuelCost}(h, r_4, c_3) = 4514$.
- The minimum contribution from r_4 with c_1 is $\text{fuelCost}(h, r_4, c_1) + \text{refuelCost}(h, r_4, c_1) - \text{fuelCost}(h, c_1) = 5776 > 4514$. c_1 can be filtered.
- The minimum contribution from r_4 with c_2 is $\text{fuelCost}(h, r_4, c_2) + \text{refuelCost}(h, r_4, c_2) - \text{fuelCost}(h, c_2) = 3245 < 4514$. c_2 cannot be filtered so optimal selection for r_4 is in $\{c_2, c_3\}$.
- All solutions are equal except for the choice of cache for r_3 and r_4 . Table 1 enumerates costs contributed by r_3 and r_4 for all combinations, so exhaustive search chooses c_3 for both r_3 and r_4 .
- The total cost is $\text{fuelCost}(h, r_1) + \text{fuelCost}(h, r_2, c_1) + 4 \times \text{fuelCost}(h, c_1) + \text{fuelCost}(h, r_3, c_3) + \text{fuelCost}(h, r_4, c_3) + \text{fuelCost}(h, c_3) = 22955$

```

ExhaustiveSearchWithFilter {
  foreach  $r_i$  in  $\{r_1, \dots, r_n\}$ 
    computeUpperBound[ $r_i$ ]
    foreach  $c_j$  in  $\{c_1, \dots, c_m\}$ 
      if (fuelCost( $h, r_i, c_j$ ) > upperBound( $r_i$ ))
        removePossibleAssignment( $r_i, c_j$ )
    examineCombinationsOfPossibleAssignments
  return optimalAssignment
}

```

Figure 5: Pseudo-code for Exhaustive Search With Filter

4. Analysis

We evaluated the methods analytically, experimentally against synthetic datasets of different sizes, and experimentally against a real world dataset provided by the Antarctica Geospatial Information Center (AGIC).

4.1 Analytical

4.1.1 Correctness

The solutions provided by the Research Flights Only and Independent Refueling Flights heuristics are guaranteed to be feasible because each starts with no selections for any research point and only makes selections where the cost is not infinite (line 5 in Figures 3 and 4). This means all selections made by the heuristic are feasible.

The solutions given by the Exhaustive Search With Filter algorithm are guaranteed to be feasible because it starts with no selections for any research point and all infeasible selections are filtered by the bounding process (line 5 in Figure 5).

4.1.2 Computational Complexity

It is easy to see that the heuristics – Research Flights Only and Independent Refueling Flights – each execute in $O(|R| \times |C|)$ time for all cases. Each possible cache site is examined once for each possible research point.

The worst case execution time for Exhaustive Search With Filter is $O(|C|^{|R|})$. Line 8 in Figure 5 is the non-polynomial part of the algorithm because it must examine all combinations of assignments. In the worst case, the filter is unable to rule out any cache site for any research point. However, if the filter is able to rule out many of the cache site choices, then the execution time can be cut down substantially. In the best case for execution time, the filter eliminates all but one choice for each research point, allowing the Exhaustive Search With Filter to execute in $O(|R| \times |C|)$ time. The average case however remains $O(|C|^{|R|})$.

4.2 Experimental Analysis

Synthetic problem instances were generated in order to compare the heuristics and the exact algorithm in terms of overall solution quality for problems with varying numbers of research points and fuel cache sites. Figure 6 shows the overall evaluation process. First random research points were selected in an area around a home point. Points within range of a direct flight from the home point were excluded. Cache sites were selected such that at least half of the research points could be reached from each cache site. Finally any research points that were not reachable from any cache site were removed and replaced with random points that were reachable from at least one cache site and not reachable from the home point directly. These restrictions on the synthetic data allowed the size of the data set to correspond to the number of alternatives the algorithm must evaluate.

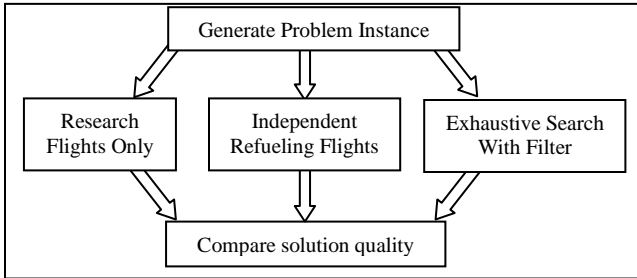


Figure 6: Experiment Design

The first experiment ran all three methods against problems with 2 cache sites and 10 – 20 research points. We ran 30 trials for each problem size. Figure 7 shows the percentage of the total cost contributed by refueling for the solutions. This is the cost divided by the total fuel required to visit all research points using the nearest fuel cache. Table 2 shows the average execution times over the trials. The run time of the exact algorithm was very volatile. The execution time for trials with 20 research points varied from 0.02 sec to 20 hours. Figure 7 and Table 2 show the Exhaustive Search With Filter provides the highest quality solutions, but it also has considerably longer execution times.

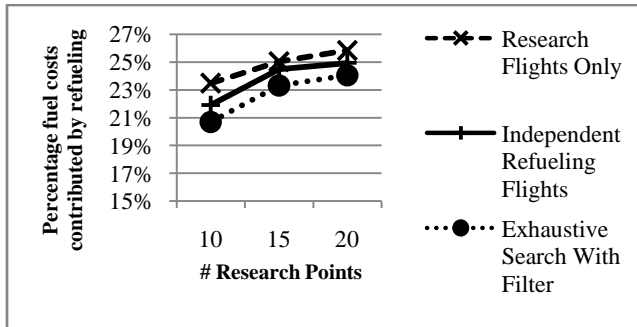


Figure 7: Percentage fuel costs contributed by refueling for problems with 2 fuel cache sites and 10 - 20 research points.

Table 2: Execution time for datasets in Figure 7

Method	Average Execution Time		
	10 points	15 points	20 points
Research Flights Only	0.001 s	0.003 s	0.005 s
Independent Refueling Flights	0.003 s	0.050 s	0.063 s
Exhaustive With Filter	0.1728 s	22.7 s	1 hour

We further evaluated the two heuristics against larger synthetic datasets in order to characterize the quality of their solutions. The first heuristic evaluation used 4 cache sites with

the number of research points varying from 10 to 100. We ran 1000 trials for each problem size. Figure 8 shows the percentage of the total cost contributed by refueling. The exhaustive search algorithm was not able to consistently finish in a reasonable time on problems of this size.

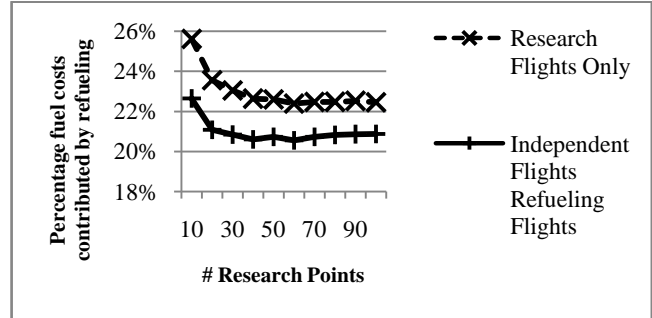


Figure 8: Percentage fuel costs contributed by refueling for problems with 4 fuel cache sites and 10 - 100 research points.

The second heuristic evaluation used 50 research points with the number of fuel cache sites varying from 2 to 10. We ran 1000 trials for each problem size. Figure 9 shows the percentage of the total cost contributed by refueling. Figure 9 shows that the Independent Refueling Flights is better able to take advantage of more fuel cache site choices than Research Flights Only. Figures 8 and 9 show that generally the Independent Refueling Flights heuristic provides a better solution. However, there were cases when the Research Flights Only heuristic provided a better solution. Both heuristics execute efficiently so it is worthwhile to try both to find solutions in an actual planning scenario.

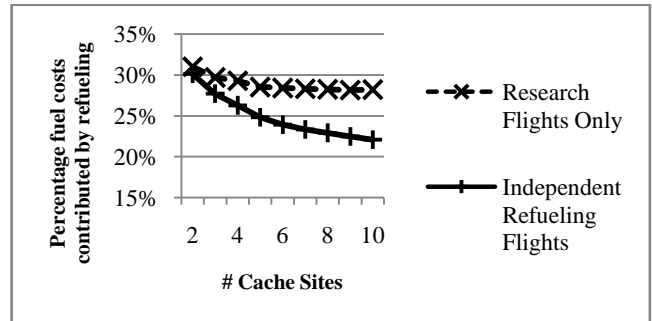


Figure 9: Percentage fuel costs contributed by refueling for problems with 60 research points and 2 - 10 fuel cache sites.

4.3 Case Study: AGIC

The Antarctica Geospatial Information Center provided a dataset for an instance of the FCSS problem. The dataset contained 26 research points, 3 existing fuel cache sites, and 2 proposed fuel cache sites. Sixteen research points were reachable by a direct flight and two were not reachable even when using the existing or proposed cache sites for refueling.

Currently AGIC evaluates proposed cache sites using two Excel spreadsheets. One spreadsheet is used to calculate the payload capacity and fuel consumption for a flight given its start, destination, and refueling stops. This method provides no clues to help the user determine which fuel cache site might be a good choice for accessing a research point. The second spread sheet is used to tally the results and keep track of the amount of fuel required at each of the cache sites.

This process suffers from several deficiencies. First, even with a given set of fuel cache sites, there is no help for choosing which cache to assign to a research point. Second, adding a new proposed cache site to the set of possible choices takes hours so only a few possible new cache sites can be evaluated. Finally, opportunities for consolidating refueling flights are missed due to the difficulty of keeping track of the solution.

An automated solution to the FCSS problem, combined with a visual representation of the problem, solves the deficiencies in the current system. For the dataset provided by AGIC, the Exhaustive Search With Filter was able to find a solution that reduced the total fuel consumption by 10% compared to the solution arrived at manually using the previous system. In addition, many more alternative cache sites can be evaluated due to the efficiency gained by solving the FCSS problem quickly. Finally, by displaying the home point, research points, and fuel cache sites in Google Earth, and presenting visual feedback on the range of research and refueling flights, the user was able to quickly identify good locations for candidate fuel caches sites and then quickly evaluate the impact of a new site on the total fuel consumption. Figure 10 shows a screenshot of the visual representation of a flight plan. The circles show the range of a round trip flight from the home point and from the fuel caches. These let a user know where it is feasible to place a fuel cache site. The lines connecting the points represent the flight path chosen by the FCSS algorithm. This is either a direct flight from the home point or a flight with a stop to refuel at a fuel cache.

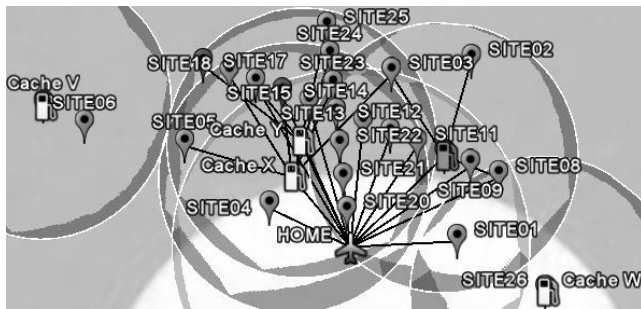


Figure 10: The problem set provided by AGIC. The circles around home and cache sites indicate the range of a flight from those sites with the given payload.

5. Conclusions and Future Work

We have characterized the fuel cache site selection (FCSS) problem and demonstrated the importance of the problem with respect to polar research and other operations in infrastructure poor regions. We proposed two heuristics and an exact algorithm. We showed through synthetic and case study data that the heuristics provide useful solutions when compared to the existing methods of cache selection, and that both the heuristics and the exact algorithm can find solutions to datasets that arise in practice.

Future areas of exploration include proving the FCSS problem is NP-hard, characterizing the problem instances where each heuristic is more effective, and identifying more effective heuristics and more efficient exact algorithms. AGIC is also interested in the creation of algorithms to recommend new cache sites given an instance of the FCSS problem.

6. ACKNOWLEDGMENTS

Special thanks to Paul Morin and Michelle LaRue from the Antarctica Geospatial Information Center and to the research

scientists in Antarctica for their help identifying and defining the FCSS problem and the software requirements. Thanks also to Kim Koffolt for improving the readability of the paper.

7. REFERENCES

- [1] Lin, S., Gertsch, N., Russell, J., 2007, A linear-time algorithm for finding optimal refueling policies. *Operations Research Letters*, 35, (3), 290-296
- [2] Lin, S., 2008. Finding optimal refueling policies: a dynamic programming approach, *Journal of Computing Sciences in Colleges*, v.23 n.6, 272-279
- [3] Wu, T.H., Low, C. and Bai, J.W. 2002. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, v. 29, 1393-1415
- [4] Avella, P., Boccia, M. and Sforza, A., Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*. v152. 170-179.
- [5] Cornillier F, Boctor FF, Laporte G and Renaud J. 2007. An exact algorithm for the petrol station replenishment problem. *Journal of the Operational Research Society*, doi:10.1057/palgrave.jors.2602374.
- [6] Giosa, I., Tansini, I. and Viera, I. 2002. New assignment algorithms for the multi-depot vehicle routing problem. *Journal of the Operational Research Society*. v53. 977-984.
- [7] Nagy, G. and Salhi, S. 2005. Heuristic algorithms for the single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*. v162. 126-141.
- [8] Renaud, J., Laporte, G. and Boctor, F. 1996. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*. V23. 229-235, doi:10.1016/0305-0548(95)00026-P
- [9] Laporte, G., Nobert, Y. and Taillefer, S. 1988. Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems. *Transportation Science*. v22. 161
- [10] Tuzun, D. and Burke, L. 1999. *European Journal of Operational Research*. v116. 87-99. doi:10.1016/S0377-2217(98)00107-6
- [11] Parthanadee, P. and Logendran, R. 2006. Periodic product distribution from multi-depots under limited supplies. *IIE Transactions*. v38. 1009 – 1026
- [12] Liu, S. and Lee, S. 2003. A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. *The International Journal of Advanced Manufacturing Technology*. v22. 941-950. doi: 10.1007/s00170-003-1639-5
- [13] Pisinger, D. and Ropke S. 2007. A general heuristic for vehicle routing problems, *Computers and Operations Research* v34. 2403–2435.
- [14] Bard, J. F., Huang, L., Jaillet, P. and Dror, M. 1998. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science* v32. 189-203.
- [15] Wu T.-H., Low C., Bai J.-W. 2002. Heuristic solutions to multi-depot location-routing problems. *Computers and Operations Research*. v29. 1393-1415. doi:10.1016/S0305-0548(01)00038-7