Generalizing Alcuin's River Crossing Problem

Michael Lampis and Valia Mitsou

School of Electrical Engineering,
National Technical University of Athens, Greece
{mlampis@cs.ntua.gr, valia@corelab.ntua.gr}

Abstract

More than 1000 years ago Alcuin of York proposed a classical puzzle involving a wolf, a goat and a bunch of cabbages that need to be ferried across a river using a boat that only has enough room for one of them. In this paper we study several generalizations of this problem, called River Crossing problems, involving more items and more complicated incompatibilities. Our study is made from an algorithmic point of view, seeking to minimize either the boat size needed, or the number of trips. We present hardness and approximation results for the case where there is no constraint on the number of trips and the incompatibilities between items are given by a general graph. We show that the same problem can be solved exactly when the graph is a tree. In addition, we present several results when there is a constraint on the number of trips and show how this variation of the problem relates to the unconstrained version.

Keywords: approximation algorithms, graph algorithms, vertex cover, transportation problems

1. Introduction

The first time algorithmic transportation problems appeared in Western literature is probably in the form of Alcuin's four "River Crossing Problems" in the book *Propositiones ad acuendos iuvenes*. Alcuin of York, who lived in the 8th century A.D. was one of the leading scholars of his time and a royal advisor in Charlemagne's court. One of Alcuin's problems was the following:

A man has to take a wolf, a goat and a bunch of cabbages across a river, but the only boat he can find has only enough room for him and one more. How can he safely transport everything to the other side, without the wolf eating the goat or the goat eating the cabbages?

This amusing problem is a very good example of a constraint satisfaction problem in operations research, and, quite surprisingly for a problem whose solution is trivial, it demonstrates many of the difficulties which are usually met when trying to solve much larger and more complicated transportation problems [Borndörfer et. Al. (1995)].

In this paper we study generalizations of Alcuin's problem which we call *River Crossing problems*. In these problems the goal is to ferry a set of items across a river, while making sure that items that remain unattended on the same bank are safe from each other. The relations between items are described by an incompatibility graph, and the objective varies from minimizing the size of the boat needed to minimizing the number of trips. Our study is made from both an algorithmic and a graph-theoretic perspective, and we seek to identify families of graphs where our problems can be efficiently solved as well as algorithms that compute exact or approximate solutions.

There are many reasons which make the study of River Crossing problems interesting and worthwhile. First, as they derive from a classic puzzle, they are amusing and entertaining, while at the same time having algorithmic depth. This makes them very valuable as a teaching tool. Several other applications of these concepts are possible. For example in cryptography, the items may represent parts of a key and the incompatibilities may indicate parts that could be combined by an adversary to gain some information. A player wishes to transfer a key to someone else, without allowing him to gain any information before the whole transaction is complete.

The rest of this paper is structured as follows: basic definitions and preliminary notions are given in Section 2. In Section 3 we study the Ferry Cover problem without constraints on the number of trips and present hardness and approximation results, as well as results for several graph topologies. In Section 4 we analyse the Trip Constrained Ferry Cover problem and present several lemmata that relate it to the unconstrained version. Section 5 consists of an analysis of the Trip Constrained Ferry Cover problem with the maximum number of trips being 3. Finally, conclusions and directions to further work are given in Section 6.

2. Definitions – Preliminaries

The rules of the River Crossing games can be roughly described as follows: we are given a set of *n* items, some of which are incompatible with each other. These incompatibilities are described in the form of a graph with vertices representing items, and edges connecting incompatible items. We need to take all *n* items across a river using a boat of fixed capacity *k* without at any point leaving two incompatible items on the same bank when the boat is not there. We seek to minimize the boat size *k* in conjunction with the number of required trips to transfer all items.

Let us now formally define the River Crossing problems we will focus on. To do this we need to define the concept of a legal configuration. Given a graph *G(V,E)*, a legal configuration is a triple (V_L, V_R, b), $V_L \cup V_R = V$, $V_L \cap V_R = \emptyset$, $b \in \{L, R\}$, s.t.

if $b = L$ then V_R induces an independent set on G else V_L induces an independent set on G. Informally, this means that when the boat is on one bank all items on the opposite bank must be compatible. Given a boat capacity k a legal left-to-right trip is a pair of legal configurations $((V_{L1}, V_{R1}, L), (V_{L2}, V_{R2}, R))$ s.t. $V_{L2} \subseteq V_{L1}$ and $|V_{L1}| - |V_{L2}| \leq k$. Similarly a right-to-left trip is a pair of legal configurations $((V_{L1}, V_{R1}, R), (V_{L2}, V_{R2}, L))$ s.t. $V_{R2} \subseteq V_{R1}$ and $|V_{R2}| - |V_{R2}| \leq k$. A ferry plan is a sequence of legal configurations starting with $(V, \emptyset, L)$ and ending with $(\emptyset, V, R)$ s.t. successive configurations constitute left-to-right or right-to-left trips.

Definition 1. The Ferry Cover (FC) problem is, given an incompatibility graph G, compute the minimum required boat size k s.t. there is a ferry plan for G.

We will denote by $\text{OPT}_{\text{FC}}(G)$ the optimal solution to the Ferry Cover problem for a graph G.

We can also define the following interesting variation of FC.

Definition 2. The Trip Constrained Ferry Cover problem is, given a graph G and an integer trip constraint m compute the minimum boat size k s.t. there is a ferry plan for G consisting of at most $2m + 2$ configurations, i.e. at most $2m + 1$ trips.

We will denote by $\text{OPT}_{\text{FC}_m}(G)$ the optimal solution of Trip Constrained Ferry Cover for a graph G given a constraint on trips m.

For the sake of completeness let us also give the definition of the well-studied Vertex Cover and Maximum Independent Set problems.

Definition 3. The Vertex Cover problem is, given a graph $G(V,E)$ find a minimum cardinality subset V' of V s.t. all edges in E have at least one endpoint in V'.

Definition 4. The Maximum Independent Set problem is, given a graph $G(V,E)$ find a maximum cardinality subset V' of V s.t. no edge in E has both endpoints in V'.

We will see that the problems defined below are closely connected to some cases of River Crossing problems.

Definition 5. The Second Vertex Cover problem is, given a graph $G(V,E)$ find an optimal vertex cover V' of G s.t. the subgraph $G'(V',E')$ induced by V' has minimum $\text{OPT}_{\text{VC}}(G')$ among all subgraphs induced by optimal vertex covers of G.

We will denote by $\text{OPT}_{\text{SVC}}(G)$ the minimum $\text{OPT}_{\text{VC}}(G')$ in the previous definition.

Definition 6. The H-Coloring problem is the following: given a fixed graph $H(V_H,E_H)$ possibly with loops but without multiple edges, we say that an input graph $G(V_G,E_G)$ has an H-Coloring if there exists a homomorphism θ from G to H, i.e. a map $\theta : V_G \rightarrow V_H$ with the property that $(u,v) \in E_G \Rightarrow (\theta(u),\theta(v)) \in E_H$.

The above problem was defined in [Hell, Nešetřil (1990)].

3. The Ferry Cover Problem

In this section we present several results for the Ferry Cover problem which indicate that it is very closely connected to Vertex Cover. We will show that Ferry Cover is NP-hard and that it has a constant factor approximation.

Lemma 1. For any graph G, $\mathrm{OPT}_{\mathrm{VC}}(G) \leq \mathrm{OPT}_{\mathrm{FC}}(G) \leq \mathrm{OPT}_{\mathrm{VC}}(G) + 1$.

Proof. For the first inequality note that if we have boat capacity k and $\mathrm{OPT}_{\mathrm{VC}}(G) > k$, then no trip is possible because any selection of k vertices to be transported on the initial trip fails to leave an independent set on the left bank.

For the second inequality, if we have boat capacity $\mathrm{OPT}_{\mathrm{VC}}(G) + 1$ then we can use the following ferry plan: load the boat with an optimal vertex cover and keep it on the boat for all the trips. Use the extra space to ferry the remaining independent set vertex by vertex to the other bank. Unload the vertex cover together with the last vertex of the independent set.

□

Theorem 1. There are constants ε_F , $n_0 > 0$ s.t. there is no $(1 + \varepsilon_F)$-approximation algorithm for Ferry Cover with instance size greater than n_0 vertices unless $P = NP$.

Proof. It is known that there is a constant $\varepsilon_S > 0$ such that there is no $(1 - \varepsilon_S)$-approximation for MAX−3SAT unless $P = NP$ [Arora et. Al. (1992)] and that there is a gap preserving reduction from MAX−3SAT to Vertex Cover. We will show that there is also a gap-preserving reduction from MAX−3SAT to Ferry Cover.

The gap-preserving reduction to Vertex Cover in [Garey, Johnson (1979)] and [Vazirani (2001)] implies that there is a constant $\varepsilon_V > 0$ s.t. for any 3CNF formula φ with m clauses we produce a graph $G(V,E)$ s.t.

$$\mathrm{OPT}_{\text{MAX-3SAT}}(\varphi) = m \Rightarrow \mathrm{OPT}_{\mathrm{VC}}(G) \leq \frac{2}{3}|V|$$

$$\mathrm{OPT}_{\text{MAX-3SAT}}(\varphi) < (1-\varepsilon_S)\cdot m \Rightarrow \mathrm{OPT}_{\mathrm{VC}}(G) > (1+\varepsilon_V)\cdot \frac{2}{3}|V|$$

In the first case it follows from Lemma 1 that

$$\mathrm{OPT}_{\mathrm{VC}}(G) \leq \frac{2}{3}|V| \Rightarrow \mathrm{OPT}_{\mathrm{FC}}(G) \leq \frac{2}{3}|V| + 1.$$

In the second case,

$$\mathrm{OPT_{VC}}(G) > (1+\varepsilon_V)\cdot\frac{2}{3}|V| \Rightarrow \mathrm{OPT_{FC}}(G) > \left(1+\varepsilon_V - \frac{1+\varepsilon_V}{\frac{2}{3}|V|+1}\right)\cdot\left(\frac{2}{3}|V|+1\right).$$

For $|V| > \frac{3}{2}\cdot\frac{1}{\varepsilon_V}$ there is a constant $\varepsilon_F > 0$ s.t. $\varepsilon_V - \frac{1+\varepsilon_V}{\frac{2}{3}|V|+1} > \varepsilon_F$.

Setting $n_0 > \left\lceil \frac{3}{2}\cdot\frac{1}{\varepsilon_V} \right\rceil$ completes the proof.

□

Corollary 1. Ferry Cover is *NP*-hard.

Proof. It follows from Theorem 1 that an algorithm which exactly solves large enough instances of Ferry Cover in polynomial time, and therefore achieves an approximation ratio better than $(1 + \varepsilon_F)$, implies that $P = NP$.

□

It should be noted that the constant ε_F in Theorem 1 is much smaller than ε_V. However, this is a consequence of using the smallest possible value for n_0. Using larger values would lead to a proof of hardness of approximation results asymptotically equivalent to those we know for Vertex Cover. This is hardly surprising, since Lemma 1 indicates that the two problems have almost equal optimum values. Lemma 1 also leads to the following approximation result for Ferry Cover.

Theorem 2. A ρ-approximation algorithm for Vertex Cover implies a $(\rho + 1/\mathrm{OPT_{FC}})$-approximation algorithm for Ferry Cover.

Proof. Consider the following algorithm: use the ρ-approximation algorithm for Vertex Cover to obtain a vertex cover of cardinality $\mathrm{SOL_{VC}}$, then set boat capacity equal to $\mathrm{SOL_{FC}} = \mathrm{SOL_{VC}} + 1$. This provides a feasible solution since loading the boat with the approximate vertex cover leaves enough room to transport the remaining independent set one by one as in Lemma 1. Observe that $\mathrm{SOL_{FC}} = \mathrm{SOL_{VC}} + 1 \leq \rho\cdot\mathrm{OPT_{VC}} + 1 \leq \rho\cdot\mathrm{OPT_{FC}} + 1$ (the first inequality from the approximation guarantee and the second from Lemma 1).

□

We now present some examples for specific graph topologies.

Example 1. If G is a clique, i.e. $G = K_n$, then $\mathrm{OPT_{FC}}(G) = \mathrm{OPT_{VC}}(G) = n - 1$.
Example 2. If G is a ring, i.e. $G = C_n$ then $\mathrm{OPT_{FC}}(G) = \mathrm{OPT_{VC}}(G) = \left\lceil \frac{n}{2} \right\rceil$.

Example 3. Consider a graph $G(V,E)$, $|V| \geq n + 3$ s.t. G contains a clique K_n and all other vertices form an independent set. In addition every vertex outside the clique is connected with every vertex of the clique. For example see Figure 1.

We will show that $OPT_{FC}(G) = OPT_{VC}(G) + 1$. Assume that $OPT_{FC}(G) = OPT_{VC}(G)$. The optimal vertex cover of G is the set of vertices of K_n. A ferry plan of G should begin by transferring the clique to the opposite bank and then leaving a vertex there. On return the only choice is to load a vertex from the independent set, because leaving any number of vertices from the clique is impossible. On arrival to the destination bank we are forced to unload the vertex from the independent set and reload the vertex from the clique. We are now at a deadlock, because none of the vertices on the boat can be unloaded on the left bank.

The graph G described in this example is a generalization of the star, where the centre vertex is replaced by a clique. The star is the simplest topology where $OPT_{FC}(G) = OPT_{VC}(G) + 1$.

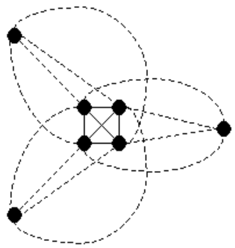

Figure 1. *An example of the graph described in Example 3*

The following theorem, together with the observation of Example 3 about stars, completely solve the Ferry Cover problem on trees.

Theorem 3. If G is a tree and $OPT_{VC}(G) \geq 2 \Rightarrow OPT_{FC}(G) = OPT_{VC}(G)$.

Proof. Let v_1, v_2 be two vertices of an optimal vertex cover of G. Then v_1 and v_2 have at most one common neighbour, because if they had at least two then G would contain a cycle. We denote by u the common neighbour of v_1 and v_2, if such a vertex exists.

Then a ferry plan of G is the following: load the vertex cover in the boat and unload v_1 in the opposite bank. Then transfer all the neighbours of v_2 vertex by vertex, leaving vertex u last to be ferried. When u is the only remaining neighbour of v_2 on the left bank, unload v_2 and load u on the boat. On arrival to the destination bank unload u and load v_1. The remaining vertices of the independent set are now transported one by one to the destination bank and finally v_2 is loaded on the boat on the last trip and transported across together with the rest of the vertex cover.

□

Remark 1. If OPT_{VC} for a tree is 1 then the tree is a star. If the star is a path then $OPT_{FC} = 1$ else $OPT_{FC} = 2$.

Corollary 2. The Ferry Cover problem can be solved in polynomial time in trees.

Proof. The Vertex Cover problem can be solved in polynomial time in trees. Theorem 3 and Remark 1 imply that determining OPT_{VC} is equivalent to determining OPT_{FC}.

4. The Trip Constrained Ferry Cover problem

Let us now focus on the trip-constrained version of our problem. First let us observe that a very tight constraint on the number of trips makes the problem trivial.

Lemma 2. For any graph *G(V,E)*, $OPT_{FC_0}(G)= |V|$

Proof. Trivial: the solution to $FC_0(G)$ is a single trip, therefore all nodes must be ferried across at once.

□

On the other hand a very loose constraint on the number of trips makes the problem equivalent to the Ferry Cover problem.

Lemma 3. For any graph *G(V,E)*, $|V| = n$, $OPT_{FC_{2^n-1}}(G) = OPT_{FC}(G)$

Proof. Any solution to FC_{2^n-1}(G) allows a ferry plan with at most 2^{n+1} configurations. There are at most 2^n legal partitions of the vertices of G into two sets, therefore there are at most 2^{n+1} possible legal configurations. No optimal ferry plan repeats the same configuration twice, since the configurations found between two successive appearances of the same configuration in a ferry plan can be omitted to produce a shorter plan. Therefore, any optimal ferry plan for the unconstrained version has length at most 2^{n+1} and can be realized within the limits of the trip constraint.

□

Loosening the trip constraint can only improve the value of the optimal solution.

Lemma 4. For any graph *G* and any integer $i \geq 0$, $OPT_{FC_i}(G) \geq OPT_{FC_{i+1}}(G)$.

Proof. Observe that a ferry plan with trip constraint *i* can also be executed with trip constraint $i + 1$.

□

A different lower bound is given by the following Lemma.

Lemma 5. For any graph *G(V,E)*, $OPT_{FC_m}(G) \geq \frac{|V|}{m+1}$

Proof. Observe that a trip constraint of *m* implies that for any ferry plan the boat will arrive at the destination bank at most $m + 1$ times. Therefore, at least one of them it must carry at least $\frac{|V|}{m+1}$ vertices.

□

Setting the trip constraint equal to the number of vertices also makes the constrained version of the problem similar to the unconstrained version.

Lemma 6. For any graph *G(V,E)*, $|V| = n$, $\text{OPT}_{\text{VC}}(G) \leq \text{OPT}_{\text{FC}_n}(G) \leq \text{OPT}_{\text{VC}}(G) + 1$.

Proof. For the first inequality, a boat capacity smaller than the minimum vertex cover allows no trips. For the second inequality it suffices to observe that the ferry plan of Lemma 1 can be realized within the trip constraint.

□

Corollary 3. Determining OPT_{FC_n} is *NP*-hard. Furthermore, there are constants ε_F, $n_0 > 0$ s.t. there is no $(1 + \varepsilon_F)$-approximation algorithm for OPT_{FC_n} with instance size greater than n_0 vertices unless $P = NP$.

Proof. Proof similar to Theorem 1, by using Lemma 6 instead of Lemma 1.

□

It is unknown whether there are graphs where $\text{OPT}_{\text{FC}_n}(G) > \text{OPT}_{\text{FC}}(G)$. We conjecture that there is a threshold $f(n)$ s.t. for any graph G, $\text{OPT}_{\text{FC}_{f(n)}}(G) = \text{OPT}_{\text{FC}}(G)$ and that $f(n)$ is much closer to n than $2^n - 1$ which was proven in Lemma 3.

5. The Trip Constrained Ferry Cover problem with trip constraint 1

An interesting special case of the Trip Constrained Ferry Cover problem is that with trip constraint *1*, i.e. the problem of computing the boat size if we allow at most *3* trips to be done. The following lemma gives a lower and an upper bound on the optimal solution of the problem.

Lemma 7. For any graph *G(V,E)*, with $|V| = n$, $\frac{n}{2} \leq \text{OPT}_{\text{FC}_1}(G) \leq n - 1$.

Proof. The first inequality is obtained by Lemma 5, where $m = 1$. For the second inequality we can use the ferry plan for a clique.

□

An other upper bound of $\text{OPT}_{\text{FC}_1}(G)$ can be obtained from the following Lemma:

Lemma 8. For any graph *G(V,E)*, $\text{OPT}_{\text{FC}_1}(G) \leq max\{\text{OPT}_{\text{VC}}(G), \text{OPT}_{\text{MAXIS}}(G) + \text{OPT}_{\text{SVC}}(G)\}$.

Proof. We can have the following ferry plan: load the optimal vertex cover of G that has the minimum optimal vertex cover on the boat. Note that the vertices that weren't loaded in the boat constitute a maximum independent set. When the boat reaches the destination bank unload as many vertices as possible i.e. $\text{OPT}_{\text{VC}}(G)$ - $\text{OPT}_{\text{SVC}}(G)$ and travel back with the remaining $\text{OPT}_{\text{SVC}}(G)$ vertices loaded. Then, for the last trip load the boat with all the vertices that were left in the first bank together with those in the boat.

To execute this ferry plan the boat needs to be of size at most max{$OPT_{VC}(G)$, $OPT_{MAXIS}(G) + OPT_{SVC}(G)$}.

□

We can also use the H-Coloring problem to obtain an equivalent definition for FC_1.

Lemma 9. A ferry plan of a graph *G* for the Trip Constrained Ferry Cover problem with constraint *1* is equivalent to an H-coloring of graph *G*, where *H* is the graph of Figure 2.

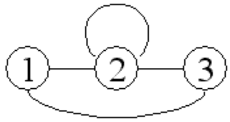

Figure 2. *Graph H of Lemma 9*

Proof. Given a ferry plan we can define the following homomorphism θ from *G* to *H*:

- $\theta(u) = 1$, where *u* is every vertex of *G* that is loaded in the first trip and is unloaded in the first trip,
- $\theta(u) = 2$, where *u* is every vertex of *G* that is loaded in the first trip and is unloaded in the third trip and
- $\theta(u) = 3$, where *u* is every vertex of *G* that is loaded in the third trip and is unloaded in the third trip.

□

Corollary 4. For any graph *G(V,E)*, $OPT_{FC_1}(G) = min\{|V_2| + max\{|V_1|,|V_3|\}\}$, where the minimum is taken among all proper H-colorings of *G* and V_1, V_2, V_3 are the subsets of *V* that have taken the colors *1*, *2* and *3* respectively.

Proof. From Lemma 9 we obtain a ferry plan for the Trip Constrained Ferry Cover problem with constraint *1*: load the subsets V_1 and V_2 in the first trip and unload the subset V_1 in the opposite bank while keeping V_2 in the boat. Then return at the first bank and load V_3 together with V_2 and transport them to the destination bank.

This implies that the boat should have room for V_2 together with one of the sets V_1 and V_3.

□

The above reformulation of $FC_1(G)$ as a problem of finding an H-coloring optimised with respect to a given function does not directly yield a result. However, we believe that it lends significant insight into the structure of the problem, and that it can be extended to cases where the trip constraint is larger than *1*. In addition, we conjecture that the above reformulation could lead to an *NP*-hardness proof for $FC_1(G)$.

6. Conclusions and Further Work

In this paper we have investigated the algorithmic complexity of several variations of River Crossing problems. For the unconstrained Ferry Cover problem we have presented results that show it is very closely related to Vertex Cover, which is a consequence of the fact that the optimal values for the two problems are almost equal. It is an open problem if, given a graph G it can be decided in polynomial time whether $OPT_{FC}(G) = OPT_{VC}(G)$ or $OPT_{FC}(G) = OPT_{VC}(G) + 1$, but we conjecture that the answer is negative.

For the Trip Constrained Ferry Cover problem, we have presented several lemmata that point out its relation to the unconstrained version. We believe that this variation is more interesting because it appears to be less related to Vertex Cover. It remains an open problem to determine at which value of the trip constraint the problem becomes equivalent to the unconstrained version (an upper bound on this value is $2^n - 1$, as shown in Lemma 3). Another interesting direction for future research is the study of the problem with fairly tight trip constraints, such as the case of FC_1 in Section 5. We conjecture that the problem remains *NP*-hard in the case of such tight trip constraints.

Acknowledgements

This paper is based on work done at the Computation and Reasoning Lab at NTUA. The authors would like to thank Aris Pagourtzis and Georgia Kaouri for the original inspiration of the algorithms exercise from which this paper began.

References

Arora S., Lund C., Motwani R., Sudan M., Szegedy M. (1992), *Proof verification and intractability of approximation problems,* In Proc. 33rd IEEE annual Symposiumon Foundations of Computer Science (FOCS), pp. 13-22.

Borndörfer R., Grötschel M., Löbel A. (1995), *Alcuin's transportation problems and integer programming.*

Garey M.R., Johnson D.S. (1979*), Computers and Intractability: A guide to the theory of NP-completeness,* W. H. Freeman and Co., New York, NY.

Hell P., Nešetřil J. (1990), *On the complexity of h-coloring,* J. Comb. Theory Ser. B, vol. 48(1), pp. 92-110.

Vazirani V.V. (2001), *Approximation Algorithms,* Spriger-Verlag Berlin Heidelberg.