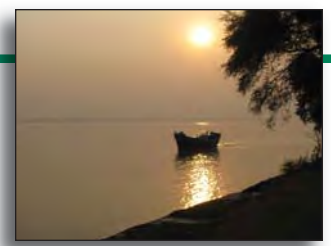


COURIER PROBLEMS

by Malay Bhattacharyya



Abstract

Courier problems comprise a set of recently proposed combinatorial optimization problems, which are inspired by some novel requirements in railway wagon scheduling. In these problems, scheduling strategies of some mobile couriers are considered. These mobile couriers initially reside at some fixed location. They are assigned some duties of commodity transfer between different pairs of locations. The scenario may be static or dynamic. The motivation is to optimize the movement of the couriers over the constraint of the traversed path or the associated cost. I discuss here some varieties of the courier problems, formalized on graphs, and address the potential methods of their solution.

Introduction

Optimization theory is one of the oldest branches of mathematics, finding a myriad of applications in the broad spectrum of scientific and engineering disciplines [1]. Many real-life problems can be formalized as an optimization problem. Even a simple problem that might arise to a fashion designer, like finding an optimal matching of tooth colors, easily finds its optimization counterpart [7]. The objective in an optimization problem is to maximize or minimize a given function $f(x)$. The nature of this function, known as the objective function, can be in any arbitrary form, e.g., univariate or multivariate, low-dimensional or high-dimensional, and so on. In single and multiple objective problems, the characteristics of the optimization function are very important. The benchmark functions to evaluate an optimization method are often guided by these features [8]. I discuss some of these characteristics of functions below.

Polynomial Function: A function given in the form $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, where a_0, a_1, \dots, a_n are constants and n is a nonnegative integer, is known as a polynomial function of order n . When $n = 1, 2, 3, 4, 5, \dots$ the function $f(x)$ is said to be linear, quadratic, cubic, quartic, quintic, and so on, respectively; e.g., $f(x) = 2x^2 + 1$ is a quadratic function.

Continuous Function: A function $f(x)$ is said to be continuous if for all the points a in its domain, $\lim_{x \rightarrow a^+} f(x) = \lim_{x \rightarrow a^-} f(x) = f(a)$. In case this relation desatisfies for any point in the domain of the function $f(x)$, it is said to be discontinuous. The function $f(x) = \sin^2 x$ is a continuous function. A function is discrete if for none of the points in its domain it is continuous; e.g., $f(x) = \lfloor x \rfloor$, denoting the smallest integer greater than or equal to x , is a discrete function.

Unimodal Function: A function $f(x)$ between two ordered sets is said to be unimodal if for some arbitrary m (the mode), it monotonically increases for $x \leq m$ and monotonically decreases for $x \geq m$, and $f(x)$ attains the only maximum value at m without attaining any other local maxima. If $f(x)$ has multiple such modes, then it is said to be a multimodal function. The function $f(x) = 10x$ is a unimodal function.

Convex Function: A real-valued function $f(x)$ defined on an interval is called convex if for any two points x and y in its domain L and for any $t \in [0, 1]$, $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$. In other words, if $f''(x) \geq 0$, for all x , then $f(x)$ is said to be convex. A function is strictly convex when the said above equality relations do not hold; e.g., $f(x) = |x|^p$ for $p \geq 1$ is a convex function.

Concave Function: A real-valued function $f(x)$ defined on an interval is called concave if for any two points x and y in its domain

L and for any $t \in [0, 1]$, $f(tx + (1-t)y) \geq tf(x) + (1-t)f(y)$. In other words, if $f''(x) \leq 0$, for all x , then $f(x)$ is said to be concave. A function is strictly concave when the said above equality relations do not hold; e.g., $f(x) = -x^2$ is a concave function.

Deterministic Function: A function is said to be deterministic if it always returns the same result as output for a specific set of input values. In contrast, if a function associates a noise term (generally additive), i.e., returns different values for the same input, it is called a stochastic function. The function $f(x) = \log x$ is a deterministic function.

Differentiable Function: A function $f(x)$ defined on an open interval is called differentiable if for any point a within the interval, the value of $\lim_{h \rightarrow 0} (f(a+h) - f(a))/h$ exists. Otherwise, $f(x)$ is non-differentiable. The function $f(x) = \tan^{-1} x$ is a differentiable function.

Optimization problems having a pronounced discrete or combinatorial structure are known as combinatorial optimization problems. Many real-life problems have a combinatorial nature. I address here a new class of combinatorial optimization problems, which are known as Courier Problems (CP). These are motivated by the complexities involved in the problem of transportation scheduling [6, 9].

Consider a system consisting of a set of locations and some mobile couriers initially residing at one of these locations. There exists a set of pairs of locations between which commodity transfer is required. These transfers of commodities involve costs that are proportional to the path distances between the corresponding pair of locations. The courier selects a pair and reaches its source location. It then collects the commodity, moves to the destination, and delivers the commodity. Again, it selects the next pair, moves to the source of the pair, and

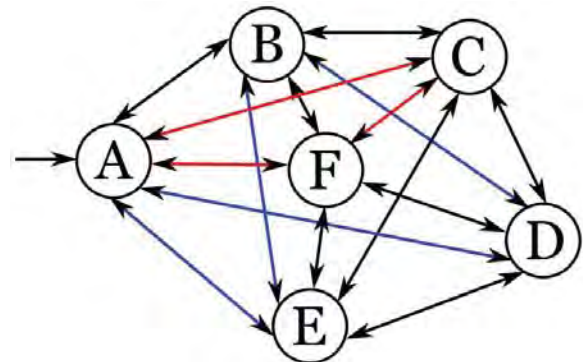


Figure 1: Scenario of a courier problem.

continues the same routine until the set of source-destination pairs is exhausted. This entire activity is termed a round. The life of a courier is defined by a sequence of many such rounds.

In a simple view, the complete scenario may be described by the four finite sets MC , L , L_C and SD , where MC denotes a finite set of couriers, $L \in L$ is finite a set of locations, L_C is the initial location of the couriers, and $SD \in L \times L$ is a set of source destination pairs. I describe one such situation in Fig. 1. Suppose there are two couriers assigned to completing the task of commodity transfer in a disjoint manner. The tuples associated with the situation in Fig. 1 are given below.

- $MC = \{MC_1, MC_2\}$,
- $L = \{A, B, C, D, E\}$,
- $L_C = A$,
- $SD = \{(A, D), (B, E), (C, F)\}$.

In this case, the two rounds shown in two different colors (red and blue) in Fig. 1 could describe a possible solution to the problem. In the following sections, similar scenarios are described with formal approach, which could possibly occur in different versions of CP.

Preliminaries of Graph Theory

A simple graph G is defined by the two finite sets V and E . The set $V = \{v_1, v_2, \dots, v_n\}$ contains the vertices and $E = \{(i, j) \mid i \neq j; i, j \in V\}$ the edges. In Fig. 1, we have $V = \{A, B, C, D, E\}$ and $E = \{V \times V - \{(A, A), (B, B), (C, C), (D, D), (E, E), (F, F)\}\}$. By considering the edges (i, j) and (j, i) dissimilar, G is assumed to be a directed graph. Here, a function W is defined over the set E , mapping the edges to a real value. According to this function, W_{ij} is the weight associated with the directed edge (i, j) . Thus, G becomes a weighted graph.

A path between two vertices v_i and v_{n+1} , denoted as $P_{v_i v_{n+1}}$, in a graph G is an ordered and alternating sequence of distinct vertices and edges $[v_i - v_2 - \dots - v_n - v_{n+1}]$, such that $v_i \in V, v_i \neq v_j$, and $(v_i, v_{i+1}) \in E, \forall i, j \in \{1, 2, \dots, n\}$. The length of this path is the number of edges it contains, i.e., n . The weight of this path $P_{v_i v_{n+1}}$ is given by $W(P_{v_i v_{n+1}}) = \sum_{i=1}^{n-1} W_{v_i v_{i+1}}$, where $W_{v_i v_{i+1}}$ denotes the weight associated with the directed edge (v_i, v_{i+1}) . A subpath is a matching subsequence of vertices and edges of a path. A circuit starting from a vertex v_i , denoted as $C_{v_i v_{n+1}}$, is a path beginning and ending with the same vertex v_i . The weight of this circuit is given by $W(C_{v_i v_{n+1}}) = \sum_{i=1}^{n-1} W_{v_i v_{i+1}} + W_{v_n v_i}$. A graph G is said to be connected if there is at least one path, having no repetition of vertices, between each vertex pair in G . In this article, the term graph will denote an arbitrary directed and weighted graph that is connected.

The Problems of Couriers

Now I formally discuss the different versions of the courier problems. A courier problem is defined over the 5-tuple (MC, L, L_C, SD, W) , where MC denotes a finite set of couriers, $L \in L$ is a finite set of locations, L_C is the initial location of the couriers, $SD \in L \times L$ is a set of source destination pairs, and W is a cost function associated with the location pairs. For the single courier problems, the tuple MC is a singleton (a set containing a single element). Therefore, it has been omitted from the appropriate cases.

Single Courier-Single Packet Problem: Given a quadruplet (L, L_C, SD, W) , find the circuit to be traversed by a courier carrying a sin-

gle commodity at a time, such that the commodity transfer between the source-destination pairs given in SD are exhaustively covered and the cost of traversal is minimum.

Single Courier-Multiple Packet Problem: Given a quadruplet (L, L_C, SD, W) , find the circuit to be traversed by a courier, allowed to carry multiple commodities at a time, such that the commodity transfer between the source-destination pairs given in SD are exhaustively covered and the cost of traversal is minimum.

Multiple Courier-Single Packet Problem: Given a 5-tuple (MC, L, L_C, SD, W) , find the $|MC|$ (the number of couriers in MC) disjoint circuits to be traversed by the couriers carrying a single commodity at a time, such that the commodity transfer between the source-destination pairs given in SD are exhaustively covered and the total cost of traversal of all the couriers is minimum.

Multiple Courier-Multiple Packet Problem: Given a 5-tuple (MC, L, L_C, SD, W) , find the $|MC|$ (the number of couriers in MC) disjoint circuits to be traversed by the couriers, allowed to carry multiple commodities at a time, such that the commodity transfer between the source-destination pairs given in SD are exhaustively covered and the total cost of traversal of all the couriers is minimum.

Similarly, the dynamic versions are defined on a dynamic environment, where the positions of the locations are changing over time. As a result, the function W , defining the cost of traversal between location pairs, also gets changed. Different versions of such transportation problems were first introduced in [3]. These problems are shown in hierarchical form in Fig. 2.

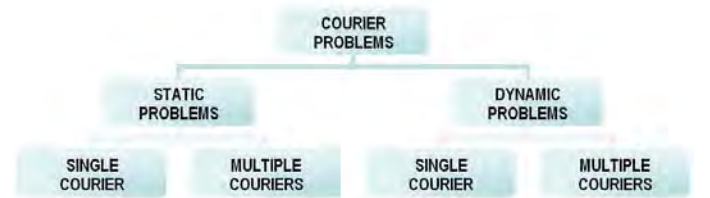


Figure 2: The problem tree of the couriers.

Approach to Solution

To date, a simplified static version of CP, concerning a single mobile courier, has only been addressed in literature [4]. It describes the first heuristic solution method of a simpler version of the single courier problem. In terminology, the problem is known as the single courier-single packet static problem or SC-SP². In brief, the problem concerns finding the route, which starts from and ends at a fixed base location amongst a given set of static locations, without repetition, for which the cost of traversal will be the minimum one. For the clarity of the solution, sometimes these problems are mapped on a graph. I also address it by mapping on a graph and, finally, to the well-known traveling salesman problem (TSP). The mapping of a single courier-single packet static problem to an equivalent graph is performed by the following rule.

- Location \rightarrow Vertex,
- Location pair \rightarrow Edge,
- Distance between a location pair \rightarrow Weight of the corresponding edge,
- Round \rightarrow Circuit.

Thus, I receive a graph for the combinatorial optimization of the circuits traversed by the couriers for minimizing the associated cost. The transformation of this graph to the TSP graph is described in the following steps.

Step 1: For every pair of locations $(l_i, l_j) \in SD$, omit the vertices corresponding to the locations l_i, l_j , and include a hypervertex $[l_i, l_j]$ in the equivalent graph.

Step 2: Connect the hypervertex $[l_i, l_j]$ with a vertex x by an edge if there was an edge (x, l_i) or (l_j, x) in the original graph, and assign the costs accordingly.

I can now solve the equivalent TSP problem on this graph. On getting the solution to this, I can find out the shortest path within the vertices of a hypervertex in the original graph and replace the hypervertices with these shortest paths. This will complete the solution circuit to be traversed by the courier.

In case of single courier-multiple packet problem, the solution approach becomes an easier one. The problem simply becomes a constrained TSP where some ordering of the locations is to be maintained. A possible solution approach to this problem is given below.

Step 1: Combine the location pairs provided in the set SD , following the rule “replace the two sets of locations (l_i, l_j) and (l_j, l_k) with a set (l_i, l_j, l_k) as appropriate.”

Step 2: Solve the TSP problem on the corresponding graph, following the constraint that the order of the vertices corresponding to the locations abide by the order of the locations derived in the modified SD .

For problems with multiple couriers, I have to sort out the disjoint circuits to be traversed by the couriers such that their total incurred cost becomes minimum. To solve such problems, I transform the problem graph into, first, a multi-salesman problem and, finally, to the standard TSP. The steps of transformation are as follows.

Step 1: For every pair of location $(l_i, l_j) \in SD$, omit the vertices corresponding to the locations l_i, l_j , and include a hypervertex $[l_i, l_j]$ in the equivalent graph.

Step 2: Connect the hypervertex $[l_i, l_j]$ with a vertex x by an edge if there was an edge (x, l_i) or (l_j, x) in the original graph, and assign the costs accordingly.

Step 3: Transform the multi-salesman problem into a standard TSP using any standard method [2].

After steps 1–2, the problem simply reduces to a multi-salesman problem. On further transformation and finally solving the TSP counterpart, I can easily get back to the original solution by retransformation; e.g., consider the problem highlighted in Fig. 1. By applying the steps 1–2, the problem gets reduced to a multi-salesman problem. The scenario is shown in Fig. 3. Now, I can suitably solve it by transforming to the standard TSP [2] and using any heuristic, such as Genetic Algorithms [5]. Suppose, I receive the solution in the form of two disjoint circuits $(A-[A, D]-[B, E]-A)$ and $(A-[C, F]-A)$ to be traversed by the two couriers. Then, the final solution will be received by replacing the hypervertices with the shortest paths in between the vertex pairs. Similarly, the multiple packet version becomes a constrained multi-salesman problem and can be solved by heuristics. Development in these directions has yet to be contributed in literature.

Conclusion

In this article, I discuss a set of novel transportation problems known as courier problems. Formal representation of courier problems on graphs

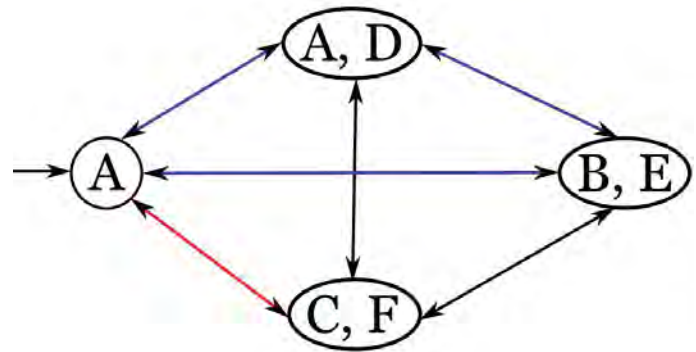


Figure 3: The scenario of a transformed multiple courier problem.

is introduced for the first time in literature. I go through some potential approaches to solutions for some of these problems. The decision versions of these problems fall in the NP-complete complexity class. Still, there remain significant open directions where a lot of progress has yet to be made in exploring more efficient solution methods, performing complexity analysis, and nurturing their behavior in dynamic environments.

References

1. Beightler, C. S., Phillips, D. T., and Wilde, D. J. 1979. *Foundations of Optimization* 2nd Ed. Prentice Hall, Englewood Cliffs, NJ.
2. Bellmore, M. and Hong, S. 1974. Transformation of multisalesmen problem to the standard traveling salesman problem. *J. ACM* 21, 3. 500-504. <http://dx.doi.org/10.1145/321832.321847>.
3. Bhattacharyya, M. 2007. Solving single courier problem using genetic Algorithms. M. E. dissertation. Jadavpur University.
4. Bhattacharyya, M. and Bandyopadhyay, A. K. 2008. On single courier problem. *Optimiz. Let.* 2, 4. 535-541. <http://dx.doi.org/10.1007/s11590-008-0079-4>.
5. Bhattacharyya, M., and Bandyopadhyay, A. K. 2009. Comparative study of some solution methods for traveling salesman problem using genetic algorithms. *Cybernetics Syst.* 40, 1. 1-24. <http://dx.doi.org/10.1080/01969720802492967>.
6. Bussieck, M., Winter, T. and Zimmermann, U. 1997. Discrete optimization in public rail transport. *Math. Programm.* 79, 1-3. 415-444. <http://dx.doi.org/10.1007/BF02614327>.
7. Cocking C. 2008. Finding an optimal tooth color match. *ACM Crossroads* 14, 3. 22-25. <http://doi.acm.org/10.1145/1373576.1373581>.
8. Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. 2007. *Evolutionary Algorithms for Solving Multi-objective Problems* 2nd Ed. Springer, Berlin, Germany.
9. Gintner, V., Kliewer, N., and Suhl, L. 2005. Solving large multi-depot multi-vehicle-type bus scheduling problems in practice. *OR Spectrum* 27, 4. 507-523. <http://dx.doi.org/10.1007/s00291-005-0207-9>.

Biography

Malay Bhattacharyya (malay_r@isical.ac.in) is a PhD scholar at the Machine Intelligence Unit of Indian Statistical Institute, Kolkata, India. He encompasses the broad domains of graph theory, logic design, soft computing, computational biology, and bioinformatics in his research. His favorite pastimes are sketching and writing songs.