Available online at www.sciencedirect.com

SciVerse ScienceDirect

ELSEVIER

Procedia - Social and Behavioral Sciences 54 (2012) 1037 – 1046

Procedia
Social and Behavioral Sciences

EWGT 2012

15th meeting of the EURO Working Group on Transportation

Persistent Visitation with Fuel Constraints

Jonathan Las Fargeas[*], Baro Hyun, Pierre Kabamba, Anouck Girard

Department of Aerospace Engineering, University of Michigan, Ann Arbor, 48109, USA

Abstract

This work is motivated by the periodic vehicle routing problem (PVRP) where a vehicle is to perpetually visit customers within a given area. In this work there is no sense of horizon or days as in classic PVRP. Instead, it is assumed that each customer has a rate at which it must be visited for the vehicle to satisfy its mission. The vehicle's fuel limitations are taken into account and fuel depots with a fixed fuel price are included. The problem of finding paths that satisfy the locations' revisit rates and minimize the total cost of fuel is treated. An algorithm that provides solutions to this problem under given constraints is presented.

Keywords: persistent visitation; fuel constraints; periodic vehicle routing problem;

1. Introduction

There are a number of customers in a given area. A vehicle is tasked with perpetually visiting these customers. Each customer must be visited at a certain rate for the vehicle to satisfy its mission. The vehicle has a finite fuel tank and consumes fuel at a given rate while in transit. Fuel depots with different fuel prices are included to allow the vehicle to refuel. The objective is to generate a path for the vehicle that satisfies the customer visitation rates and minimizes the cost of fuel.

1.1. Literature review

The problem stated here is based on the Vehicle Routing Problem (VRP) but is also similar to patrol problems. Thus research addressing these topics is reviewed.

[*] Corresponding author.
E-mail address: jfargeas@umich.edu.

 doi:10.1016/j.sbspro.2012.09.819

The vehicle routing problem and methods to solve it are introduced in (Christofides, 1976). In VRP, a vehicle is tasked with performing pickup and delivery operations from a depot to a set of customers in a given area; the goal is to find a path for the vehicle that minimizes a given metric whether it be time, distance, or cost. The Period Vehicle Routing Problem (PVRP) is presented in (Christofides & Beasley, 1984) where a multiple day horizon is introduced and some customers must be visited on several days at a given frequency. In (Desrochers, Lenstra, Savelsbergh, & Soumis, 1988) the Vehicle Routing Problem with Time Windows (VRPTW) is studied, in this VRP variation each customer must be visited within a certain window of time. A heuristic to solve the periodic vehicle routing problem with time windows, which is a combination of PVRP and VRPTW, is presented in (Cordeau, Laporte, & Mercier, 2001). The emissions VRP is introduced in (Figliozzi, 2010) where the goal is to minimize the emissions and fuel consumption of the vehicle.

When the pickup and delivery aspects of VRP are removed, the problem is reduced to the Traveling-Salesman Problem (TSP). In TSP the goal is to find the shortest tour for a salesman starting from a given location, visiting each of a specified group of locations, and then returning to the departure point. For TSP, exact algorithms suffer from computational complexity that increases as a function of the number of visiting locations. Instead, heuristics provide good solutions within reasonable time, but they do not guarantee finding the optimum. For existing TSP heuristics, see (Oberlin, Rathina, & Darbha, 2010).

The patrolling problem consists of continuously visiting, with one or more agents, locations of interest in an area such that the time between visits to the same location is minimized. Heuristics to solve the patrolling problem under given conditions are presented in (Wolfer Calvo & Cordone, 2003). The patrolling problem with multiple agents is treated in (Chevaleyre, 2004). The formulation and solution of a patrolling problem with incidents occurring with a known distribution in time and space where the goal is to minimize the expected wait time between the occurrence of an incident and its detection are described in (Huynh, Enright, & Frazzoli, 2010).

In (Basilico, Gatti, & Amigoni, 2009) a method for a patrolling agent to combat intrusion is given. The authors define the problem using a graph, where each node is a potential target and has an intrusion time indicating how much time it will take an intruder to break into the node. The method derived results in the patrolling agent visiting the nodes perpetually where the time between consecutive visits for a given node is less than its intrusion time. This work is extended to multiple patrolling agents in (Basilico, Gatti, & Villa, 2010).

While the current literature examines many variations of VRP and patrolling problems, no method implements continuous visitation frequencies and accounts for the fuel costs of the vehicle. The current paper addresses this subject.

1.2. Original contributions

- Formulation of minimization of fuel cost for persistent visitation with fuel constraints problems.
- Proof of existence of periodic solutions.
- Complete algorithm to solve for tours that minimize fuel cost.

1.3. Relevance to past work

Previously, we posed the persistent visitation problem without fuel constraints where a mobile agent was tasked with visiting objects of interest perpetually while satisfying heterogeneous revisit deadlines for the objects. We investigated the periodicity properties of solutions and presented incomplete heuristics to solve the problem (Las Fargeas, Hyun, Kabamba, & Girard, 2012).

1.4. Paper outline

The remainder of the paper is as follows. In Section 2, the modeling for the revisit deadlines of the customers and the refueling process are presented. In Section 3, important characteristics of the vehicle path under given conditions are derived. In Section 4, the problem is formulated. In Section 5, we present the algorithm to find tours that, if infinitely repeated, solve the problem. Conclusions are discussed in Section 6.

2. Modeling

In this problem, operations occur in a two dimensional area with a single vehicle, n customers, and q fuel depots. The vehicle is assumed to be traveling at a constant velocity v. The n customers each have Cartesian coordinates (ξ_i, ζ_i) and a finite positive revisit deadline $r_i \in \mathbb{R}, 1 \leq i \leq n$. The vehicle has a finite fuel capacity of F and consumes fuel at a finite constant rate of $\dot{f}_c$ per unit time where $\dot{f}_c \geq 0$, the vehicle endurance is thus $r_f = (F/\dot{f}_c)$. The q fuel depots have Cartesian coordinates (ξ_i, ζ_i) and a finite constant positive fuel cost $c_i \in \mathbb{R}, n < i \leq n+q$. The fuel depots store an infinite amount of fuel. The vehicle can select how much fuel to purchase when visiting a fuel depot. Visits and refueling are assumed to have negligible durations in comparison to the time spent in transit.

2.1. Revisit deadlines

In classic PVRP, planning is done over a finite number of days and customers have a frequency of visitation (per unit day). The PVRP formulation allows for the decoupling of scheduling, which customers to visit on which days, and the vehicle path planning, what path to take to visit a given set of customers on a certain day.

In this work, there is no sense of days and visits are done continuously. The frequency of visitation for a customer is per unit time. The revisit deadline is the inverse of the frequency of visitation. The vehicle cannot bank time between visits to the same customer, the finite revisit deadline indicates that the time between consecutive visits to customer i must be less than or equal to r_i. This work does not include the pickup and delivery of goods aspect of VRP; in this regard it is more similar to TSP.

2.2. State space model

The following states are introduced: $p(k) \in \{1,2,\dots,n+q\}$ indicates which customer or depot is being visited upon completion of step k, where a step is the act of the vehicle traveling from one location or depot to the next, $\tau(k) \in \mathbb{R}$ indicates the total time elapsed upon completion of step k. Note that this discretization of the vehicle movements in steps works because there is only one vehicle operating; if there were multiple vehicles the steps would not be synchronized and another methodology would have to be used, likely including the position and heading of the vehicles. Let $y(k) \in \{1,2,\dots,n+q\} \times \mathbb{R}$ contain the following discrete time states:

$$y(k) = [p(k) \;\; \tau(k)\,]^T. \tag{1}$$

The continuous time states are now introduced: $x_i(t) \in \mathbb{R}, 1 \leq i \leq n$ is the slack time of customer i, which indicates how much longer the vehicle can wait before a visit to customer i is overdue and $f(t) \in \mathbb{R}$ is the amount of fuel the vehicle is carrying at time $t \in \mathbb{R}$. The mission state $m(t) \in \mathbb{R}^{n+1}$ contains the following continuous states:

$$m(t) = [x_1(t) \;\; x_2(t) \;\; \dots \;\; x_n(t) \;\; f(t)]^T. \tag{2}$$

The input used is $u(k) \in \{1,2,\dots,n+q\} \times \mathbb{R}$, it contains which customer or depot the vehicle will visit next and how much fuel the vehicle will purchase if the next visit is a depot visit (k indicates the step at which this input is applied and $l \in \{1,2\}$ where $l = 1$ indicates that $u(k,l) \in \{1,2,\dots,n+q\}$ corresponds to the visiting location and $l = 2$ indicates that $u(k,l) \in \mathbb{R}$ corresponds to the fuel amount being purchased). Fuel can only be purchased at fuel depots thus:

$$u(k,1) \leq n \Rightarrow u(k,2) = 0. \tag{1}$$

Note that the discrete time states for step $k+1$ are determined by the discrete time states and the input for step k. Also, note that the time rate of change of the mission state at time t is a function of the time t, the customer or the depot being visited upon completion of step k, and the history of the amounts of fuel purchased. Considering this, the following model is used:

$$y(k+1) = f(y(k), u(k,1)), \tag{2}$$
$$\dot{m}(t) = g(t, y_{1\le i\le k}(i), u_{1\le i\le k}(i,2)). \tag{3}$$

2.3. Initial conditions

The initial conditions of the state space variables are as follows,

$$\begin{gathered} p(0) = u(1,1), \\ \tau(0) = 0, \\ x_1(0) = r_1, \\ x_2(0) = r_2, \\ \vdots \\ x_n(0) = r_n, \\ f(0) = f_0. \end{gathered} \tag{4}$$

The amount of fuel the vehicle carries initially f_0 is an additional input to the model.

2.4. Dynamics

The dynamics of the visitation schedule p and the total time elapsed τ are as follows:

$$p(k+1) = u(k,1), \tag{5}$$
$$u(k,1) \neq p(k), \tag{6}$$
$$\tau(k+1) = \tau(k) + \sqrt{(\xi_{u(k)} - \xi_{p(k)})^2 + (\zeta_{u(k)} - \zeta_{p(k)})^2}/v. \tag{7}$$

The dynamics of the slack time for customer i is:

$$\dot{x}_i(t) = -1 + \sum_{j=1}^{k} \delta_{ip(j)} \cdot \left(r_i - x_i(\tau(j))\right) \cdot \delta\left(t - \tau(j)\right), \;\; t \le \tau(k), 1 \le i \le n, \tag{8}$$

where $\delta(t - \tau(j))$ is the continuous Dirac delta function and $\delta_{ip(j)}$ is the discrete Kronecker delta function. The dynamics of the fuel the vehicle is carrying is given as,

$$\dot{f}(t) = -\dot{f}_c + \sum_{j=1}^{k} \mathrm{u}(j,2) \cdot \delta\left(t - \tau(j)\right), \qquad t \le \tau(k). \tag{9}$$

3. Vehicle path characteristics

3.1. Periodicity

A *path* is defined as an infinite sequence of visitations such that no customer visitation is ever overdue and the vehicle's fuel is always bounded by zero and the fuel tank size.

Definition 3.1. *A tour is defined as a sequence of visitations starting from one customer and ending at that same customer such that all other customers have been visited.*

A path can thus be expressed as an infinite sequence of tours. The following remark and lemmas are used to prove the existence of periodic tours within paths.

Remark 3.2. *Viewing a path as an infinite sequence of tours, the initial mission state of these tours belongs to the compact set* $[0, r_1] \times [0, r_2] \times \ldots \times [0, r_n] \times [0, r_f]$.

Lemma 3.3. *If a path exists, then a path with a periodic mission state exists.*

Proof: If a path exists, then an infinite sequence of tours exists. Each tour has an initial mission state; hence if a path exists there exists an infinite sequence of initial mission states of tours. The initial mission state of tours belongs to a compact set (Remark 3.2), thus the sequence of initial mission states of tours is bounded. The Bolzano-Weierstrass Theorem states that *every bounded sequence has a convergent subsequence*. Hence a convergent subsequence can be extracted from the infinite sequence of initial mission states of tours. Let τ_i be the i^{th} tour in the path, thus

$$\forall \varepsilon \in \mathbb{R}^{n+1} > 0, \exists N(\varepsilon) \in \mathbb{N}\ such\ that\ \forall i \geq N(\varepsilon), \left\| m_{final_{\tau_i}} - m_{final_{\tau_{i+1}}} \right\| < \varepsilon \tag{10}$$

Thus a path can be constructed consisting of infinite repetitions of the tour τ_i where $i \geq N$. Therefore, a path with periodic mission states exists.

Lemma 3.4. *If no depot or customer is equidistant to two other depots, then there exists a unique sequence of visitations describing the mission state on* $[t_0, t_1]$ *where at least one visit to a customer occurs in* $[t_0, t_1]$.

Proof: Impulses in a slack time in the mission state are uniquely described by visits to the customer the slack time represents. Thus visits to customers can be extracted from the impulses slack times in the mission state, this is called labeling. Thus all slack time impulses can be labeled directly. Impulses in the amount of fuel in the mission state are not uniquely described by visits to a certain fuel depot as multiple depots exist thus they cannot be uniquely labeled directly. Because no customer is equidistant to two depots, the time in transit from any customer to a depot is unique. Thus the time between an impulse in a slack time and an adjacent impulse in the amount of fuel uniquely determines which depot visit caused the impulse in the amount of fuel. Thus impulses in the amount of fuel adjacent to an impulse in a slack time can be labeled. No depot is equidistant to two other depots, thus the time in transit from a depot to another is unique. Therefore, an impulse in the amount of fuel adjacent to a labeled impulse in the amount of fuel can be labeled. Hence all impulses occurring in the elements of the mission state on a given time interval can be labeled if at least one slack time impulse occurs within the time interval.

Theorem 3.5. *If a path exists and no depot or customer is equidistant to two other depots, then a periodic path exists.*

Proof: If a path exists then according to Lemma 3.3 a path with a periodic mission state exists. Let $P(t)$ be the mission state during a period. In $P(t)$ at least one visit to a customer occurs since for a slack time to return to a higher value an impulse must occur, and no depot or customer is equidistant to two other depots thus according to Lemma 3.4 $P(t)$ is described by a unique sequence of visitations. Therefore, the sequence of visitations of the path with periodic mission state is periodic.

Depots can be moved by infinitesimal amounts to circumvent the equidistance issue and guarantee the existence of tours.

3.2. Finite number of tours

Theorem 3.6. *There exists a finite number of tours that start at customers which if infinitely repeated can solve the persistent visitation problem.*

Proof: The time in transit from one location to another is finite, thus a finite sequence of visitations takes finite time to complete. The duration of a tour that starts at a customer is constrained by the revisit deadline of the starting customer. There are a finite number of combinations of visits that can be achieved within a finite time interval. Thus the number of tours starting from a given customer that can solve the problem is finite. There are a finite number of customers, hence there exist a finite number of tours that start at customers and solve the persistent visitation problem if infinitely repeated.

4. Problem formulation

The goal of the agent in the persistent visitation problem with fuel constraints is to find the tour that minimizes the cost per unit time while satisfying the fuel and slack time constraints. Thus, the tour needs to satisfy continuity constraints for the slack times and the fuel. The time in transit between consecutive visits to a customer must be less than or equal to that customer's revisit deadline. The fuel consumed in transit between consecutive visits to fuel depots must be less than or equal to the vehicle fuel capacity. In addition, infinite repetitions of this tour must form a solution. Thus, the tour must satisfy closure properties. The time in transit between the last visit and the first visit to a customer must be less than that customer's revisit deadline and the fuel consumed between the last depot visit and the first depot visit must be less than the vehicle fuel capacity.

Let $T(k), k \in \mathbb{N}$, be a tour of length L. Let O be the matrix of visitation indices, O has $n+1$ rows where the first n rows correspond to visits to customers and the last row corresponds to visits to fuel depots. $O_i(k)$ is the index of the k^{th} visit to customer i in the tour and $O_{n+1}(k)$ is the index of the k^{th} visit to a fuel depot in the tour. Let l_i be the length of O_i, l_{n+1} be the length of O_{n+1}. For example, if $O(2)$ contains $\{2,5,7\}$ then customer 2 was visited at steps 2, 5, and 7, equivalently $T(2)=T(5)=T(7)=2$. If there are four customers, then $O(5)=\{3,4,6\}$ means that fuel depots were visited at steps 3, 4, and 6, equivalently $T(3)>4$, $T(4)>4$, and $T(6)>4$ (since there are four customers the fuel depots are represented by integers higher than 4). Let $h(i,j)$ be the distance between location i and location j.

Based on the model above, we formulate the problem as follows: the vehicle is to find a tour T such that the total fuel cost per unit time is minimized and the following conditions on continuity and closure of slack times and fuel hold:

$$\begin{cases} \sum_{j=o_i(k)}^{o_i(k+1)-1} h(T(j),T(j+1))/v \le r_i, 1 \le i \le n, 1 \le k \le l_i - 1 \\ \sum_{j=o_{n+1}(k)}^{o_{n+1}(k+1)-1} h(T(j),T(j+1))/v \le r_f, 1 \le k \le l_i - 1 \\ \sum_{j=o_i(l_i)}^{L-1} h(T(j),T(j+1))/v + \sum_{j=1}^{o_i(1)-1} h(T(j),T(j+1))/v \le r_i, 1 \le i \le n \\ \sum_{j=o_{n+1}(l_{n+1})}^{L-1} h(T(j),T(j+1))/v + \sum_{j=1}^{o_{n+1}(1)-1} h(T(j),T(j+1))/v \le r_f \end{cases} \tag{11}$$

The first two equations are the continuity constraints and the last two equations are the closure constraints. The slack time continuity and closure constraints are equivalent to the intrusion time constraints used in (Basilico, Gatti, & Amigoni, 2009).

5. Problem solution

5.1. Main Algorithm structure

The main algorithm to solve this problem is composed of multiple parts. The first part of the main algorithm (Algorithm 1) finds all tours that satisfy the slack time constraints and can satisfy the fuel constraint without solving for the amounts of fuel to be purchased; it uses the functions presented in Algorithm 2 and Algorithm 3 for constraint verification. The second part of the main algorithm solves a constrained minimization problem for each tour to calculate the amounts of fuel to be purchased that satisfy the fuel constraint and minimize the total cost spent on fuel during the tour. The third part of the main algorithm selects the tour with the minimum total cost divided by the time length, thus choosing the tour with the minimum cost per unit time.

Π=findTours(T,O,Π)
1. *L←length(T)*
2. **for** *i←1:(n+q)*
3. **if** *L*=0 and *i* > *n* **then go to step 2**
4. **if** *L* ≥ 1 and *i=T(last)* **then go to step 2**
5. *T'←T.add(i)*
6. *O'←O*
7. **if** *i* ≤ n **then** *O'(i) ←O'(i).add(L+1)* **else** *O'(n+1) ←O'(n+1).add(L+1)*
8. **if** *missionStateContinuity(T',O')* **then** *a←true* **else** *a←false*
9. **if** *T'(1)=T'(last)* **then** *b←true* **else** *b←false*
10. **if** *b* and *missionStateClosure(T',O')* **then** *c←true* **else** *c←false*
11. **if** *a* and *b* and *c* **then** add *T'* to *Π*
12. **else if** *a* and $\bar{b}$ **then** *Π=findTours(T',O',Π)*
13. **end**
14. **return** *Π*

Algorithm 1. Algorithm to find all valid tours starting from customers.

5.1.1. Algorithm to find tours

This algorithm finds all tours that can solve the problem and start from customers. The algorithm uses three main variables in its computation: *T* is the sequence of visitations of the current tour, *O* is the matrix of visitation indices, and *Π* is the set of valid tours.

missionStateContinuity(T,O)
1. *L←length(T)*
2. *K←length(O(n+1))*
3. **if** K=0 **then** v_1, v_2 ←1 **else if** K=1 **then** v_1←1,v_2 ←*O(n+1)* **else** v_1, v_2 ← last two entries in *O(n+1)*
3. $t \leftarrow \sum_{i=v_1}^{v_2-1} locations(T(i)).distance(locations(T(i+1)))/v$
4. **if** $t > r_f$ **then return false**
5. $t \leftarrow \sum_{i=v_2}^{L-1} locations(T(i).distance(locations(T(i+1)))/v$
6. **if** $t > r_f$ **then return false**
7. **for** i←*1:n*
8. *K←length(O(i))*
9. **if** K=0 **then** v_1, v_2 ←1 **else if** K=1 **then** v_1←1,v_2 ←*O(i)* **else** v_1, v_2 ← last two entries in *O(i)*
10. $t \leftarrow \sum_{i=v_1}^{v_2-1} locations(T(i)).distance(locations(T(i+1)))/v$
11. **if** $t > r_i$ **then return false**
12. $t \leftarrow \sum_{i=v_2}^{L-1} locations(T(i)).distance(locations(T(i+1)))/v$
13. **if** $t > r_i$ **then return false**
14. **end**
15. **return true**

Algorithm 2. Function to verify continuity of mission states.

The recursive algorithm used is presented in Algorithm 1. The algorithm iterates through the possible locations to be visited, appends a visit to the current tour, and checks whether the current sequence satisfies the continuity and closure conditions. Specific steps proceed as follows; steps 2 through 13 are the loop for potential next visits. Step 3 forces tours to start at customers and step 4 does not allow consecutive visits to the same location. Step 7 adds an index to the matrix of visitation indices for the new visit. Step 8 checks the continuity conditions of the mission state, step 9 checks whether the tour has completed, and step 10 checks the closure conditions of the mission state if the tour has completed. In step 11, if all conditions are satisfied then the tour is added to the set of valid tours, while if the tour satisfies the continuity conditions and is incomplete the search on that tour is continued as indicated in step 12, otherwise the tour is rejected.

Algorithm 1 calls two other functions to check the continuity and closure conditions of the mission state. These functions implement the restrictions in Equation *(11)* and are presented in Algorithm 2 and Algorithm 3 respectively.

The mission state continuity function, Algorithm 2, first verifies the continuity conditions for the fuel and then verifies the continuity conditions for all the slack times. From one iteration in Algorithm 1 to the next, the length of the sequence can be changed by zero or one thus there is no need to verify that all continuity constraints in Equation *(11)* are satisfied; instead this function verifies the last two visits to customers and depots. The fuel continuity verification is achieved by checking that the time between the last two depot visits and the time between the last depot visit and the current step are less than the vehicle endurance. The slack time continuity verification is achieved by checking that the slack time for customer *i* between the last two visits to customer *i* and the time elapsed from the last visit to customer *i* and the current step are less than the revisit deadline for customer *i*.

missionStateClosure(T,O)
$L \leftarrow length(T)$
$v_1 \leftarrow$ last entry in $O(n+1)$, $v_2 \leftarrow$ first entry in $O(n+1)$
$t \leftarrow \sum_{i=v_1}^{L-1} locations(T(i)).distance(locations(T(i+1)))/v$
$t \leftarrow t + \sum_{i=1}^{v_2-1} locations(T(i)).distance(locations(T(i+1)))/v$
if $t > F/\dot{f}_c$ **then return false**
for $i \leftarrow 1:n$
    $v_1 \leftarrow$ last entry in $O(i)$, $v_2 \leftarrow$ first entry in $O(i)$
    $t \leftarrow \sum_{i=v_1}^{L-1} locations(T(i)).distance(locations(T(i+1)))/v$
    $t \leftarrow t + \sum_{i=1}^{v_2-1} locations(T(i)).distance(locations(T(i+1)))/v$
    if $t > r_i$ **then return false**
end
return true

Algorithm 3. Function to verify closure of mission states.

Similarly, the mission state closure function, Algorithm 3, checks that the time elapsed between the last fuel depot visit and the first is less than the vehicle endurance and that the time elapsed between the last visit to customer *i* and the first visit to customer *i* is less than the revisit deadline for customer *i*.

5.1.2. Tour fuel cost minimization

For each tour, a constrained minimization problem is solved to calculate the fuel to be purchased at each depot such that the total cost of fuel is minimized. Let *T* be a tour, the sequence of fuel depot visits, *D*, can be extracted from *T*: $D = \{d_1, d_2, \ldots, d_m\}$ where d_i is the i^{th} depot visited. We define $\Delta t_i, 1 \leq i \leq m-1$ as the time spent traveling between d_i and d_{i+1}, Δt_m is the time in transit between d_m and d_1. Let Δf_i be the amount of fuel purchased by the vehicle when visiting d_i.

During the tour, the fuel the vehicle is carrying must always be bounded by zero and the vehicle fuel capacity. Thus the amount of fuel purchased at a given step must be large enough for the vehicle to reach the next depot but small enough as to not surpass the fuel capacity. This requirement results in the following constraints on the amount of fuel purchased:

$$\begin{gathered} \dot{f}_c \Delta t_1 \leq \Delta f_1 \leq F, \\ \dot{f}_c (\Delta t_1 + \Delta t_2) \leq \Delta f_1 + \Delta f_2 \leq F + \dot{f}_c \Delta t_1, \\ \vdots \\ \dot{f}_c \sum_{i=1}^{m} \Delta t_i \leq \sum_{i=1}^{m} \Delta f_i \leq F + \dot{f}_c \sum_{i=1}^{m-1} \Delta t_i . \end{gathered} \tag{12}$$

These constraints can be recasted in the following matrix inequality form:

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_m \end{bmatrix} \leq \begin{bmatrix} -\dot{f}_c \Delta t_1 \\ -\dot{f}_c (\Delta t_1 + \Delta t_2) \\ \vdots \\ -\dot{f}_c \sum_{i=1}^{m} \Delta t_i \\ F \\ F + \dot{f}_c \Delta t_1 \\ \vdots \\ F + \dot{f}_c \sum_{i=1}^{m-1} \Delta t_i \end{bmatrix}. \quad (13)$$

In addition, the amount of fuel purchased must always be greater than or equal to zero and less than or equal to the fuel capacity:

$$0 \leq \Delta f_i \leq F, 1 \leq i \leq m. \quad (14)$$

The goal of the vehicle is to minimize the cost per unit time, since the duration of the tour is already fixed the time aspect can be ignored thus the minimization is expressed as follows:

$$\min_{\Delta f_i, 1 \leq i \leq m} \sum_{j=1}^{m} \Delta f_j c_{d_j}. \quad (15)$$

Equations *(13)*, *(14)*, and *(15)* form a constrained minimization problem which can be solved using linear programming methods.

5.1.3. Minimum cost per unit time tour

The third part of the algorithm selects the tour that will result in the path with minimum cost per unit time:

$$T_{best} = \min_{T \in \Pi} \left(\frac{\sum_{j=1}^{m(T)} \Delta f_j(T) c_{d_j(T)}}{\tau(L(T))} \right). \quad (16)$$

The initial conditions for the path generated by infinite repetitions of a tour are as stated in 2.3 where $f_0 = f(\tau(L(T_{best})))$.

5.2. Algorithm correctness, completeness, and complexity

The tour finding algorithm verifies that slack time and fuel continuity are satisfied whenever a visitation is added; in addition slack time and fuel closure are verified before a tour is admitted, and thus the tour finding portion of the algorithm is correct. The tour fuel cost minimization portion of the algorithm ensures that the amount of fuel purchased at each depot is such that the amount of fuel the vehicle is carrying is always positive and less than or equal to the fuel capacity. Hence the tour fuel cost minimization portion of the algorithm is correct. Therefore, the algorithm is correct.

The tour finding algorithm searches for all possible combinations of visits within the search depth set by the revisit deadline of the starting customer and returns tours that satisfy the slack time and fuel constraints. As such, the algorithm is complete by exhaustion. Because of this search methodology, the algorithm's performance at best is of the order $(n + q - 1)^{n+q}$. However, the slack time and fuel constraints mean that in reality the algorithm will reject a sequence of visitations that violates a constraint before searching the maximum depth.

The algorithm can be extended to allow tours to start at depots as well as customers however a finite number of tours would no longer be guaranteed thus additional constraints to guarantee termination might be required. In addition, if this extension were implemented the algorithm would no longer be complete since there would be no limit to the length of solutions.

6. Conclusions

In this paper, a formulation of a periodic vehicle routing problem with continuous revisit deadlines and allowing fuel purchases is presented. Properties of paths that solve the problem are derived. A complete algorithm to find tours starting at customers which solve the problem is given.

Acknowledgements

The research was supported in part by the United States Air Force grant FA 8650-07-2-3744.

References

Basilico, N., Gatti, N., & Amigoni, F. (2009). Developing a deterministic patrolling strategy for security agents. *Proceedings of theIEEE/WIC/ACM International Joint Conferences on Web Intelligenceand Intelligent Agent Technologies*, *2*, pp. 565-572.

Basilico, N., Gatti, N., & Villa, F. (2010). Asynchronous multi-robot patrolling against intrusions in arbitrary topologies. *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, (pp. 1224-1229).

Chevaleyre, Y. (2004). Theoretical Analysis of the multi-agent patrolling problem. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, (pp. 302-308).

Christofides, N. (1976, February). The Vehicle Routing Problem. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle, 10*(2), 55-70.

Christofides, N., & Beasley, J. (1984). The period routing problem. *Networks, 14*, 237-256.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society, 52*, 928-936.

Desrochers, M., Lenstra, J., Savelsbergh, M., & Soumis, F. (1988). Vehicle routing with time windows: Optimization and approximation. In E. S. Publishers, *Vehicle Routing: Methods and Studies* (pp. 65-84).

Figliozzi, M. (2010). Vehicle Routing Problem for Emissions Minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 1-7.

Huynh, V., Enright, J., & Frazzoli, E. (2010). Persistent patrol with limited range on-board sensors. *Proceedings of the 49th IEEE Conference on Decision and Control*, (pp. 7661-7668).

Las Fargeas, J., Hyun, B., Kabamba, P., & Girard, A. (2012). Persistent Visitation with Heterogeneous Revisit Rate Requirements. *Manuscript submitted for publication.*

Oberlin, P., Rathina, S., & Darbha, S. (2010, December). Today's Traveling Salesman Problem. *IEEE Robotics Automation Magazine, 17*(4), 70-77.

Wolfer Calvo, R., & Cordone, R. (2003). A heuristic approach to the overnight security service problem. *Computers & Operations Research, 30*(9), 1269-1287.