JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **4**, Number **1**, February **2008**

Website: http://AIMsciences.org

pp. **1–XX**

COMPUTATIONAL ASPECTS OF THE OPTIMAL TRANSIT PATH PROBLEM

Louis Caccetta, Ian Loosen and Volker Rehbock

Western Australian Centre of Excellence in Industrial Optimisation
Department of Mathematics and Statistics
Curtin University of Technology
Bentley, WA, 6102, Australia

Abstract. We consider a class of path design problems which arise when an object needs to traverse between two points through a specified region. The path must optimise a prescribed criterion such as risk, reliability or cost and satisfy a number of constraints such as total travel time. Problems of this type readily arise in the defence, transport and communication industries. We specifically look at the problem of determining an optimal (in terms of minimizing the overall probability of detection) transit path for a submarine moving through a field of sonar sensors, subject to a total time constraint. A computational strategy along with results are presented.

1. **Introduction.** A fundamental network design problem is that of designing "efficient" routes for moving products, resources and information through a network. Usually the route must satisfy a variety of constraints. The efficiency of a network can be measured in terms of cost, reliability, throughput or length of path used. The constraints of a network can consist of delivery/pickup time deadlines, network availability, the need to visit specific nodes of the network, or resource restrictions such as vehicle capacity, fleet size or transmission rates to name a few.

We consider a class of path design problems which arise when an object needs to traverse between two points through a specified region, the Transit Path Problem being one such problem. The Transit Path Problem is to determine an optimal path, in terms of minimizing risk or cost or maximizing reliability, for an object, such as a robot or vehicle, that needs to traverse a specified region between two points. This problem arises in many areas of real life. For example, the routing of military vehicles through a detection field or the routing of a new highway in a given terrain. Other applications include motion planning for robot manipulators through a field of obstacles and the generation of optimal trajectories for air, space, naval and land vehicles. We specifically look at the problem of determining an optimal transit path for a submarine moving through a field of sonar sensors, subject to a total time constraint.

The strategy presented involves a two stage approach. The first stage is a discretization of the problem and the development of a network heuristic method to solve the resulting network. The second stage involves the use of an optimal control model and a solution procedure that utilizes the solution obtained in the first stage

2000 *Mathematics Subject Classification.* Primary: 49J15, 49N90; Secondary: 90B10.
Key words and phrases. Optimal Control, optimisation, discretization, transit path.
This work was supported by an Australian Research Council Grant.

as a starting point. In this phase of our procedure we make use of the optimal control software package MISER3.

In the proposed model each of the sensors can detect the presence of the submarine with a probability which is a given function depending on the distance between the two and on the speed of the submarine. This function is not a simple analytical expression, but depends upon a range of factors, including the characteristics of the ocean floor and ocean surface, depths of the sensor and the submarine, and the temperature and salinity of the water.

In this paper, we use probability of detection functions reported in Hallam [12]. These were constructed under the assumptions that the geographic location and environmental conditions are known and that the submarine remains at a constant depth. Furthermore, each of the given functions is constructed for a particular constant vessel speed. While there are still further factors influencing the probability of detection (such as machinery states, frequency of the sensor, alertness of sensor operators or quality of the automatic detection, the relative aspect of the submarine and the sensor, the effect of sudden changes in travel direction or speed), the functions from Hallam [12] display sufficient complexity to test the feasibility of the proposed method. The overall probability of detection at any point in time can then be calculated as an appropriate combination of these individual probabilities of detection. Here, we make the assumption that the probability of detection for any one sensor is independent of the probabilities of detection for the other sensors. The objective then is to find a transit path between two fixed positions in the sensor field which will minimize the overall probability of detection while still satisfying a maximum travel time constraint. The difficulty is due to the fact that the transit time must satisfy an upper bound constraint.

In an earlier paper [5] we addressed the same transit path problem with the objective of testing the efficiency of various non-linear optimization routines used within MISER3 for the optimal control phase of our procedure. In this paper we investigate the effects of different degrees of coarseness of discretization used in the problem. In particular, we show how the subsequent number of switching points needed within the optimal control phase can make a significant difference on the final solution obtained.

This paper is organized as follows. In Section 2 we describe the general optimal control formulation. The CPET technique is explained in Section 3. Section 4 presents the computational strategy we used to solve the problem. In Section 5 we give numerical results. Finally we summarize our conclusions in Section 6.

2. **Optimal Control Formulation.**

The optimal control formulation follows the treatment given in Rehbock *et al.* [23]. A sonar field is positioned in the Cartesian plane with coordinates (x, y) indicating the latitudinal and longitudinal distance (in kms) from the origin. Letting $(x(t), y(t))$ represent the location of the submarine at time t, the system's dynamics is described by:

$$\begin{aligned} \dot{x}(t) &= s(t)\cos(\theta(t)), \quad x(0) = 0, \\ \dot{y}(t) &= s(t)\sin(\theta(t)), \quad y(0) = 0, \end{aligned} \tag{1}$$

where $\theta(t)$ represents the heading angle of the vessel in radians and $s(t)$ is the speed of the vessel in km/h. Note that θ and s are control functions satisfying:

$$0 \leq \theta(t) \leq 2\pi, \ \forall t \in [0, T), \text{ and} \tag{2}$$

$$s(t) \in \{s_1, s_2, s_3, ..., \} \ \forall t \in [0, T). \tag{3}$$

Suppose that a total of n_s sensors are located at positions (x_i, y_i), $i = 1, 2, ..., n_s$, in the field. We assume for simplicity that these positions remain fixed during the journey and that the sonars are all of the same type with the same detection capabilities. At any instant of time, the distance of the submarine from each sensor is given by $r_i(t) = \sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2}$, $i = 1$, ..., n_s. For a vessel speed s, a probability of detection profile, $p(r, s)$ can be constructed as a function of the physical distance $r(t)$. Assuming that the sensors operate independently, the instantaneous probability of the vessel being detected is then given by:

$$P((x(t), y(t), s(t)) = 1 - \prod_{i=1}^{n_s} (1 - p_i(r_i(t), s(t))). \tag{4}$$

Our aim is to minimize the cumulative probability of being detected over the entire journey. This is equivalent to minimizing the objective functional:

$$g(\theta, s, T) = \int_0^T P((x(t), y(t), s(t))dt. \tag{5}$$

For the submarine to arrive at its intended destination within a prescribed total time, we have constraints:

$$x(T) = x_T, \ y(T) = y_T, \tag{6}$$

$$T \leq T_{MAX}. \tag{7}$$

Note that the terminal time, T, is variable in this problem. In summary, then, the optimal control model of the submarine transit path problem can be stated as: *Find a terminal time T (satisfying (7)), and control functions $\theta(t)$ (satisfying (2)) and $s(t)$ (satisfying (3)) such that the objective functional (5) is minimized subject to the vessel dynamics (1) and the constraints (6).*

The fact that the control s is restricted to a discrete set of values places this problem into a general class of discrete valued optimal control problems. Examples of these problems are studied in Howlett *et al.* [15] and Lee *et al.* [20]. The main difficulty with these problems is to determine the exact time points where the discrete valued control should switch between its allowed values. Since the gradients with respect to these switching time parameters are discontinuous, ordinary gradient based solution methods perform poorly. An additional difficulty is to determine exactly how many such switching times are involved in an optimal solution. The first of these difficulties has been successfully overcome by the *Control Parameterization Enhancing Transform* (CPET), which was initially applied to a similar class of time optimal control problems in Lee *et al.* [19] and later directly to discrete valued optimal control problems in Lee *et al.* [20]. The second difficulty can be partially addressed by solving a sequence of problems which are transformed via CPET, but this remains an active area of research. Essentially, CPET involves a scaling of the time horizon, $[0, T]$, via an auxiliary control function known as the *enhancing control*. This transforms the original problem into an equivalent canonical form which can then be solved by ordinary gradient based methods such as control parameterization described in Teo *et al.* [27] and incorporated into the optimal control software MISER3 [17].

3. **CPET Technique.** We briefly explain CPET through its application to the example at hand. For a more thorough review and discussion of these techniques, see Rehbock *et al.* [22]. Our first task is to set a limit to the number of course/speed switchings to be allowed and the allowable speeds s. Note that the heading angle

control function, $\theta(t)$, is modelled as a piecewise constant function, which is natural, given that the heading angle ought to remain constant between course changes. Furthermore, for the sake of simplicity, we assume that the switching times for the course changes coincide with switching times for the speed changes. This may appear to be restrictive, but note that this formulation does allow for only one of the controls to change value at a particular switching time, so full generality of the control structure is actually preserved.

In our computations, the submarine is restricted to two speeds, 8 km/h and 14 km/h. The control constraint (3), therefore, becomes:

$$s(t) \in \{8, 14\}, \ \forall t \in [0, T). \tag{8}$$

The detection profiles are given in the form of a set of data points and cubic splines are used to interpolate this data to generate smooth $p(r, s)$ curves [5].

In this application, we have a practical limitation on the number of course/speed changes during the time horizon, because course and speed changes physically require a minimum period of time to be implemented. Furthermore, a submarine commander is unlikely to implement a solution which involves an excessive number of course/speed changes. Hence, we assume that the maximum number of switchings allowed is $N - 1$. The CPET technique may then be applied as follows.

We define a new time horizon $[0, N]$ and partition it into the subintervals $I_1 = [0, 1)$, $I_2 = [1, 2)$, $I_3 = [2, 3)$, ..., $I_N = [N - 1, N)$. We then define $u_1(\tau), \tau \in [0, N)$ to be a piecewise constant function on $[0, N)$ that is consistent with this partition. u_1 is essentially the heading angle of the submarine in the transformed time scale and we still require the control constraints:

$$0 \leq u_1(\tau) \leq 2\pi, \ \forall \tau \in [0, N). \tag{9}$$

Furthermore, we define

$$u_2(\tau) = \begin{cases} 14, & \text{if } \tau \in I_k, \quad k \text{ odd}, \\ 8, & \text{if } \tau \in I_k, \quad k \text{ even}. \end{cases} \tag{10}$$

This (fixed) control function takes on the role of $s(t)$ in the transformed problem. Note that it is consistent with the constraint (8). Furthermore, we define the *enhancing control*, $u_3(\tau)$, to be a piecewise constant function consistent with the above partition and subject to the following constraints:

$$0 \leq u_3(\tau) \ \leq \ T_{MAX}. \tag{11}$$

The constraint (11) arises due to the total time constraint (7), but, by itself, will not be sufficient to replace (7) entirely. The main feature of the CPET method is the scaling, via the enhancing control, which relates the original time horizon $[0, T]$ to the new time horizon $[0, N]$. This is done through the following differential equation:

$$\frac{dt}{d\tau} = u_3(\tau), \ \tau \in [0, N), \ t(0) = 0. \tag{12}$$

Note that integration of (12) over $[0, N)$ will allow us to recover the original time horizon $[0, T]$, where $T = t(N)$. To standardize notation, we set, $x_1 = x$, $x_2 = y$ and $x_3 = t$. The transformed problem may then be stated as follows. Find control functions $u_1(\tau)$ and $u_3(\tau)$ such that:

$$\int_0^N P(x_1(\tau), x_2(\tau), u_2(\tau)) u_3(\tau) d\tau, \tag{13}$$

is minimized subject to the dynamics

$$\begin{array}{ll} \dot{x_1}(\tau) = u_3(\tau)u_2(\tau)\cos(u_1(\tau)), & x_1(0) = 0, \\ \dot{x_2}(\tau) = u_3(\tau)u_2(\tau)\sin(u_1(\tau)), & x_2(0) = 0, \\ \dot{x_3}(\tau) = u_3, & x_3(0) = 0, \end{array} \tag{14}$$

and the constraints

$$\begin{array}{l} x_1(N) = x_T, \\ x_2(N) = y_T, \\ x_3(N) \leq T_{MAX}. \end{array} \tag{15}$$

Note that the third constraint in (15) arises directly from (7).

The transformed problem now simply involves piecewise constant control functions defined on a regular fixed partition of the fixed time horizon $[0, N]$. As such, it can be solved directly by the optimal control software MISER3 [18]. Note that the optimal solution of the original problem can be recovered easily from the solution of the transformed problem, as the original time scale is given by $x_3(\tau)$.

4. **Computational Strategy.** The Transit Path Problem is likely to have a unique optimal solution. However, in the solution region there are typically a large number of peaks and troughs representative of local maxima and minima solutions. While a locally optimal solution may be easily calculated, past numerical experiences [23] suggests it is very much dependent on the initial guess one provides. It is unlikely that, given an arbitrary initial guess, a direct optimal control solution of the problem will generate a path which is globally optimal. To generate a good initial path we solve a discretized approximation of the problem by first constructing a grid-like network over the sensor field, then solve the resulting Constrained Shortest Path Problem (CSPP). We regard the knot points of the grid as nodes and the grid lines as edges, and thus think of the grid as a graph. We allow movement along edges only and with each edge we can associate a cost value. The cost depends on the location of the edge in the sensor field and speed at which the vessel travels along the edge. It is calculated in the same manner as for the optimal control model. Furthermore, since the distance along each edge and the travel speed are known, we can calculate the time it takes the vessel to traverse each edge.

In the optimal control formulation, we noted that the speed s of the submarine can come from a finite discrete set (3). Therefore, for each of the speeds in the discrete set, we need to have a corresponding edge between the node i and j in the graph to represent each of the speeds $s_1, s_2, s_3, \ldots$. Assuming that the speeds in the set (3) are arranged in increasing order, $s_1 < s_2 < s_3 < ...$, it is possible to perform some elementary preprocessing. Letting the cost along the edge associated with speed s_n be denoted by c_{ij}^n, we can clearly eliminate the edge (i, j) if $c_{ij}^n \geq min\{c_{ij}^m\}$ $\forall\ m > n$.

Next we introduce dummy nodes and edges so as to avoid multiple edges. Of those possible speeds that remain after preprocessing, we can leave the edge (i, j) joining the slowest speed from node i to j. For each of the remaining speeds we need a dummy node j' and a dummy edge (j', j), which has a zero cost and a zero transit time.

With respect to the simple graph $G = (N, A)$, where $N = \{1, 2, ..., n\}$ is the set of nodes, $|N| = n$, and A is the set of edges, each edge (i, j) has a corresponding cost c_{ij} and a transit time t_{ij}. For convenience, we denote the start (origin) node by $'O'$ and the destination node by $'D'$. Also, let T_{MAX} be the time limit. The CSPP can be expressed as an Integer Programming Problem [1]. The CSPP has been

studied by a number of authors and both exact algorithms and heuristics have been proposed. Methods used to solve CSPP include: k-shortest path ideas ([3],[28]), cost scaling and rounding ([10], [14], [21]), Dynamic Programming formulations ([8], [16]), and label setting approaches ([2], [3], [10]). By using the Integer Programming Formulation, solution strategies are: Lagrangian relaxation methods ([6], [13], [26]), Branch and Bound algorithms [24], Branch and Cut methods [4], or to apply a linear integer programming package such as CPLEX [7].

The CSPP is a computationally difficult problem to solve, because it is NP-complete. For the purpose of evaluating our strategy we generate a feasible path using a simple heuristic that finds a solution which is near optimal or optimal with minimal computational time. Basically, each edge (i, j) in our graph has two weights associated with it: cost c_{ij} and traverse time t_{ij}. The idea is to parameterize these two weights into one label on the edge using the convex combination $w_{ij} = \alpha c_{ij} + (1-\alpha)t_{ij}$, where $\alpha \in [0, 1]$. Then, we solve the resulting SPP, using Dijkstra's algorithm [9], while incrementing α until a time feasible path, $t(P) \leq T_{MAX}$, is found. This path forms our initial guess for the optimal control phase of our approach. It is envisaged that this path is reasonably close to the global optimal solution for the continuous optimal control problem and, therefore, should be sufficient for our purpose.

5. **Computational Results.** All the computational tests were carried out on two separate machines. Our network heuristic was tested on a Sun Netra X1 with a 500MHz 64-bit Ultra SPRAC Processor and 256MB of RAM. Whereas the optimal control model results were obtained via a Intel Pentium 4 PC with a 2.40GHz Processor and 512MB of RAM.

Our approach uses the solutions obtained from our heuristic as initial starting points for the corresponding optimal control model. The MISER3 software was used in conjunction with several different nonlinear programming solvers (NLPQL [25], FFSQP [29], & NPSOL [11]) to refine these initial solutions. FFSQP is based on the concept of feasible sequential quadratic programming. Starting with a feasible point (provided by the user or generated automatically), the algorithm produces successive iterates that all satisfy the constraints. The objective function can be decreased either after each iteration with an Armijo-type arc search or after at most three iterations with a nonmonotone line search. The user has the option to choose one of the two searches. The merit function used in both searches is the objective function itself. NLPQL is also based on a sequential quadratic programming method. Working with a quadratic approximation of the Lagrangian function and a linearization of the constraints, a quadratic subproblem is formulated and solved by the dual code QL. Following this, a line search is performed with respect to two alternative merit functions and the Hessian approximation is updated by the modified BFGS-formula. Unlike FFSQP, a feasibilty requirement is not imposed on the iterates. Like NLPQL, NPSOL is also a sequential quadratic programming method incorporating an augmented Lagrangian merit function and a BFGS quasi-Newton approximation to the Hessian of the Lagrangian.

To test our methodology, we ran two sets of test problems. Our first set of problems allows us to not only test our approach but also compare the different nonlinear solvers within MISER3. We generated our second set of problems so as to measure the sensitivity of the optimal control phase subject to the initial grid size employed. We tested our proposed method and the three different nonlinear solvers on 480 problems that were generated as follows.

We imposed 4 different grid sizes over the region for the network phase, these being of dimension 20×20, 40×40, 60×60 and 80×80, which are all equally spaced rectangular grids. We have restricted our test problems to sonar fields with $n_s = 4$ sensors, though any number can be easily incorporated. Furthermore, we chose to look at a region of 80 km by 80 km. In addition, $x_0 = 0$, $y_0 = 0$, $x_T = 80$ and $y_T = 80$, that is the starting point is (0, 0) km and the destination is at the point (80, 80) km. We then constructed 30 different sets of sensor locations. The locations of the sensors were determined by randomly generating 240 integer values, $\{x_1, x_2, ..., x_{240}\}$, between 0 and 80. We then paired these together, $(x_1, x_2), (x_3, x_4), ..., (x_{239}, x_{240})$, to give the coordinates, in kilometres, of the sensors in relation to the starting point of the journey. The first four pairs give the sensor locations for the first set, the next four pairs represent the sensor locations for the second set, and so on. We use the probability of detection curves given in Hallam [12].

For each grid dimension and set of sensor locations, we imposed four different time constraints T_{MAX}, ranging from a low or "tight" time constraint to a high or "loose" time constraint. For each problem, we determined its minimum time path, denoted by T_L, and the time corresponding to the minimum unconstrained shortest path, denoted as T_H. The four time constraints are then found using the formulae $T_\alpha = (1 - \alpha)T_L + \alpha T_H$ for $\alpha = 0.2$, 0.4, 0.6 and 0.8. $T_{0.2}$ represents the "tightest" constraint and $T_{0.8}$ the "loosest" constraint.

Recall that the optimal control model does not require the path to move along the grid lines. Instead, virtually any concatenation of straight line sections with any direction and one of the two possible speeds is allowed. Clearly, the optimal control model is less constrained than the network approximation, so we expect to see an improvement in the optimal cost obtained.

From the first set of test problems, we were able to establish that each of the three optimisation routines has its own advantage. NPSOL gives the best average percentage improvement, NLPQL has the quickest CPU time and FFSQP is the most "accurate" in terms of meeting the optimal control constraints. It appears that there is a three way trade-off between objective function value, CPU time and accuracy. It is up to the user to use the optimisation routine that is most suited to his or her needs. For a thorough presentation and discussion of these results refer to Caccetta *et al.* [5].

To test the effect that the grid size has on the final solution, we generated 120 problems. To do this, we used the same set of sensor locations as in our previous computational tests. The four different "degrees" of tightness were also generated in the same manner. However, we imposed the time constraint associated with the 20×20 network to the other three grid dimensions. In addition, we only used the NPSOL nonlinear programming solver for these test problems. By maintaining the same time constraint for all the grid sizes, we are able to directly compare the effect that the initial starting solutions generated via the different sizes has on the final path obtained in the optimal control phase. We employed the NPSOL solver for this because it has the greatest average percentage improvement, its CPU time is reasonable and its "accuracy" is comparable to the FFSQP solver.

In Table 1, we present the average number of switching points used within the optimal control phase. A switching point is needed whenever the submarine makes a course or speed change. We notice from Table 1 that as the grid size increases so does the amount of switches required. The reason being that larger grid sizes have more

edges associated with them therefore there is more opportunity for the submarine to make a course or speed change than with a smaller grid size. The amount of switches was always kept to a minimum, that is, we only used a sufficient amount that allowed us to incorporate the heuristic solution into the required MISER input format.

	20×20	40×40	60×60	80×80
# Switches	17.64167	27.80833	39.375	49.425

TABLE 1. Average number of switching points.

In Table 2, we present the results for all the grid sizes. Each table displays the average computational results for all 120 problems broken up into the cases $\alpha = 0.2, 0.4, 0.6, 0.8$ as well as the total average over all of the cases. We present the percentage improvement achieved by use of the optimisation routine NPSOL, when compared to the initial starting path. Also shown is the average computational time, in minutes, taken to generate the solution by the NPSOL optimisation routine.

	20×20		40×40		60×60		80×80	
α	% Imp.	CPU Time	% Imp.	CPU Time	% Imp.	CPU Time	% Imp.	CPU Time
0.2	31.24	1.80	30.70	2.89	30.94	3.38	31.49	3.21
0.4	26.13	2.55	25.32	3.42	24.47	3.31	23.01	4.30
0.6	17.59	3.01	16.76	3.27	17.59	4.66	16.67	4.66
0.8	11.77	3.31	12.57	3.55	10.97	4.70	11.26	8.14
All	21.68	2.67	21.34	3.28	20.99	4.01	20.61	5.08

TABLE 2. Computational results for the comparison of different grid sizes.

As we observed with our earlier results [5], Table 2 shows us that the percentage improvement that is made over the initial solution decreases as the grid size increases, the reason being that as we increase the grid size the solution generated via the heuristic will be more closely representative of a continuous solution. Therefore the change from the heuristic discrete solution to the optimal control continuous solution wont be as dramatic as it would be for a smaller grid size. This is what results in the lower average % improvement we see in Table 2. We also note that as the grid dimension becomes larger the CPU time, in minutes, increases. The reason being, as pointed out earlier, larger grid dimensions have more switches associated with them. These switches increase the complexity of the optimal control problem and thereby the CPU time required to solve it.

For both the heuristic and optimal control solution we measured the average percentage of how far the results obtained were from the corresponding best solution of that approach. For example, for the heuristic phase, after determining the solution for each grid size using the same time constraint, we compared the costs of these paths $C(P)$ against the best $C(P^*)$ of the four solutions using the equation $[(C(P) - C(P^*))/C(P^*)] \times 100$. This procedure was also followed for the

	20×20	40×40	60×60	80×80
0.2	2.83	2.25	1.87	1.57
0.4	3.02	2.39	1.43	1.40
0.6	2.68	2.33	1.92	1.42
0.8	4.04	2.09	0.83	0.20
All	3.14	2.27	1.51	1.15

TABLE 3. Heuristic: average % above best solution.

optimal control solutions. This tells us how far, in terms of percentage above the best solution, the path was from the best solution obtained.

From Table 3, we see that, as the grid size increases from 20×20 to 80×80, the quality of the Heuristic solution generated improves from 3.14% to 1.15% on average above the best solution. However, when we look at Table 4 we note that the best solution obtained via the optimal control phase comes from the starting solution generated using the 60×60 grid. It is on average 4.56% above the best solution. The worst solution, on average, was found when using the 20×20 path which was 5.40% greater than the best solution.

	20×20	40×40	60×60	80×80
0.2	6.03	6.26	5.15	4.32
0.4	4.35	4.88	5.05	7.07
0.6	5.74	6.53	4.89	5.65
0.8	5.48	2.55	3.16	2.25
All	5.40	5.05	4.56	4.82

TABLE 4. Optimal Control: average % above best solution.

This shows us that despite using a "better" path from the 80×80 grid dimension as an initial guess the resulting optimal control path is not as good as the path resulting from using the 60×60 grid starting solution. This contradicts our belief that the best optimal control solution would be found by using a feasible path which has the lowest cost. These results and observations are a direct consequence of the different number of switching points used within the optimal control phase. We still believe that the final optimal control solution is dependent on a good initial solution. However, the number of switching points also has a large bearing on the quality of result obtained. This would appear to be due to the nature of the CPET transformation which tends to introduce a large number of local minima in the transformed optimal control formulation of the problem. Consequently the likelihood of getting stuck in a local minima increases with the number of switching points.

6. **Conclusions.** We have considered the Transit Path Problem, and in particular, the problem of finding an optimal submarine transit path through a field of sonar sensors. Our approach to solving the problem was to use two phases. First, we generate a good quality solution for the discretized network problem, using a simple efficient heuristic. Then, we refine it by means of an optimal control approach. Our

work seems to suggest that the number of switches employed is as an important factor in determining a good solution as is the quality of the initial solution used for the optimal control phase.

Acknowledgements. This work was supported by an Australian Research Council Grant (No: DP0346396). The authors would like to also thank Robert O'Dowd and Christina Hallam from the Maritime Operations Division in the Defence Science and Technology Organisation for their input through discussions and the provision of test data.

REFERENCES

[1] R.L. Ahuja, J.L. Magnanti and J.B. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice-Hall, Englewood Cliff, New Jersey, 1993.
[2] Y.P. Aneja, V. Aggarwal and K.P.K. Nair, *Shortest Chain Subject to Side Constraints*, Networks, **13** (1983), no. 2, 295–302.
[3] J.E. Beasley and N. Christofides, *An Algorithm for the Resource Constrained Shortest Path Problem*, Networks, **19** (1989), no. 4, 379–394.
[4] L. Caccetta, *Branch and Cut Methods for Mixed Integer Programming Problems*, Progess in Optimization II, Kluwer Publishers, Applied Optimization Series, **39** (2000), 21–44.
[5] L. Caccetta, I. Loosen and V. Rehbock, *Optimal transit path problem for submarines*, Modelling Supplementary Volume of the DCDIS, Series B, Applications & Algorithms, (2005), 874–880.
[6] W.M. Carlyle and R.K. Wood, *Lagrangian Relaxation and Enumeration for Solving Constrained Shortest-Path Problems*, Proceedings of the 38th Annual ORSNZ Conference, University of Waikato, (2003), 3–12.
[7] *CPLEX 6.0 User Manual*, ILOG Inc., CPLEX Division, Incline Village, NV, USA, (1997).
[8] M. Desrochers and F. Soumis, *A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows*, INFOR, **26** (1988), 191–212.
[9] E.W. Dijkstra, *A note of two problems in connection with graphs*, Numerische Mathematik, **1** (1959), 269–271.
[10] I. Dumitrescu and N. Boland, *Improved Preprocessing, Labeling and Scaling Algorithms for the Weight Constrained Shortest Path Problem*, Networks, **42** (2003), no. 3, 135–153.
[11] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, *User's Guide for NPSOL (Version 4.0): A Fortran package for nonlinear programming*, Report SOL 86-2,Department of Operations Research, Stanford University, (1986).
[12] C.L. Hallam, *Hierarchical Path Generation: An Application to Submarine Transit Paths*, Honours Dissertation, Murdoch University, Western Australia, (1997).
[13] G. Handler and I. Zang, *A Dual Algorithm for the Constrained Shortest Path Problem*, Networks, **10** (1980), no. 4, 293–310.
[14] R. Hassin, *Approximation Schemes for the Restricted Shortest Path Problem*, Mathematics of Operations Research, **17** (1992), 36–42.
[15] P. Howlett, P. Pudney and B. Benjamin, *Determination of optimal driving strategies for the control of a train*, in Computational Techniques and Applications: CTAC-91, Computational Mathematics Group, Australian Mathematical Society, (1991), 241–248.
[16] B. Jaumard, F. Semet and T. Vovor, *A Two-Phase Resource Constrained Shortest Path Algorithm for Acyclic Graphs*, Les Cahier du GERAD, (1996), G-96-48.
[17] L.S. Jennings, M.E. Fisher, K.L. Teo and C.J. Goh, *MISER3: Solving Optimal Control Problems - An Update*, Advanced Engineering Software, **13** (1991), 190–196.
[18] L.S. Jennings, K.L. Teo, M.E. Fisher and C.J. Goh, *MISER3 version 2, Optimal Control Software, Theory and User Manual*, http://cado.maths.uwa.edu.au/miser/manual.html, Department of Mathematics, University of Western Australia.
[19] H.W.J. Lee, K.L. Teo, L.S. Jennings and V. Rehbock, *Control Parametrization Enhancing Technique for Time Optimal Control Problems*, Dynamic Systems and Applications, **6** (1997), no. 2, 243–262.
[20] H.W.J. Lee, K.L. Teo, V. Rehbock and L.S. Jennings, *Control Parametrization Enhancing Technique for Discrete-Valued Control Problems*, Automatica, **35** (1999), no. 8, 1401–1407.

[21] D.H. Lorenz and D. Raz, *A Simple Efficient Approximation Scheme for the Restricted Shortest Path Problem*, Operations Research Letters, **28** (2001), 213–219.

[22] V. Rehbock, K.L. Teo, L.S. Jennings and H.W.J. Lee, *A Survey of the Control Parametrization and Control Parametrization Enhancing Methods for Constrained Optimal Control Problems*, Progress in Optimization: Contributions from Australia, Kluwer Academic Press, (1999), 247–275.

[23] V. Rehbock, L. Caccetta, C.L. Hallam and R. O'Dowd, *Optimal Submarine Transit Paths Through Sonar Fields*, Research Report, Department of Mathematics and Statistics, Curtin University of Technology, Western Australia, (2000).

[24] C. Ribeiro and M. Minoux, *Solving Hard Constrained Shortest Path Problems by Lagrangian Relaxation and Branch-and-Bound Algorithms*, Methods of Operations Research, **53** (1986), 303–316.

[25] K. Schittkowski, *NLPQL: A Fortran subroutine for solving constrained nonlinear programming problems*, Annals of Operations Research, **5** (1985/86), 485–500.

[26] J.F. Shapiro, *A Survey of Lagrangian Techniques for Discrete Optimization*, Annals of Discrete Mathematics, **5** (1979), 113–138.

[27] K.L. Teo, C.J. Goh and K.H. Wong, *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific and Technical, London, (1991).

[28] J.Y. Yen, *Finding the k-Shortest, Loopless Paths in a Network*, Management Science, **17** (1971), 711–715.

[29] J.L. Zhou and A.L. Tits, *A Users Guide for FFSQP Version 3.5: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfyin All Inequality Constraints*, Institute for Systems Research, University of Maryland, (1995).

Received September 2006; revised November 2006

E-mail address: caccetta@maths.curtin.edu.au
E-mail address: ian.loosen@postgrad.curtin.edu.au
E-mail address: rehbock@maths.curtin.edu.au