Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Multi-Goal Path Optimization for Robotic Systems with Redundancy based on the Traveling Salesman Problem with Neighborhoods

PRESENTED BY Iacopo Gentilini

ACCEPTED BY THE DEPARTMENT OF

Mechanical Engineering

Kenji Shimada 2012.06.04

ADVISOR, MAJOR PROFESSOR DATE

Nadine Aubry 06/04/2012

DEPARTMENT HEAD DATE

APPROVED BY THE COLLEGE COUNCIL

Vijayakumar Bhagavatula July 06, 2012

DEAN DATE

Multi-Goal Path Optimization for Robotic Systems with Redundancy based on the Traveling Salesman Problem with Neighborhoods

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Mechanical Engineering

Iacopo Gentilini

Laurea, Mechanical Engineering, Politecnico di Torino, Italy
Diplom, Mechanical Engineering, Universität Karlsruhe (TH), Germany
M.S., Mechanical Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May, 2012

Acknowledgments

The author would like to express his deep gratitude and appreciation to his advisor and chair of the doctoral committee, Dr. Kenji Shimada, and to the other member of the committee, Dr. François Margot, Dr. William C. Messner, and Dr. David Bourne.

He also would like to acknowledge DENSO Wave, Inc, Japan, and the Autodesk IDEA Studio program for providing enlightening ideas, the test environments, and partial funding for this research.

Finally, a special thanks is directed to the Bertucci Graduate Fellowship in Engineering and the Mary Jane and Milton C. Shaw Fellowship for their encouraging trust and financial support.

Abstract

Finding an optimal path for a redundant robotic system to visit a sequence of several goal locations is a complex optimization problem and poses two main technical challenges. Because of the redundancy in the system, the robot can assume an infinite number of goal configurations to reach each goal location. Therefore, not only an optimal sequence of the goals has to be defined, but also, for each goal, an optimal configuration has to be chosen among infinite possibilities. Second, the actual cost for the system to move from one configuration to the next depends on many factors, such as obstacle avoidance or energy consumption, and can be calculated only through the employment of specific path planning techniques.

We first address the optimization problem of finding an optimal sequence of optimal configurations, while assuming the cost function to be analytically defined. This problem is modeled as a Traveling Salesman Problem with Neighborhoods (TSPN), which extends the well-known TSP to more general cases where each vertex (goal configuration) is allowed to move in a given region (neighborhood). In the literature, heuristic solution approaches are available for TSPN instances with only circular or spherical neighborhoods. For more general neighborhood topologies, but limited to the Euclidean norm as edge weighting function, approximation algorithms have also been proposed. We present three novel approaches: (1) a global Mixed Integer Non Linear Programming (MINLP) optimizer and (2) a convex MINLP optimizer are modified to solve to optimality TSPN instances with up to 20 convex neighborhoods, and (3) a hybrid random-key Genetic Algorithm (GA) is developed to address more general problems with a larger number of possibly non-convex neighborhoods and with different types of edge weighting functions. Benchmark tests show that the GA is able to find the same optimal tour calculated by the MINLP solvers while drastically reducing the computational cost, and it always improves the best known solutions for available test problems with up to 1,000 neighborhoods.

Second, we integrate the GA with a probabilistic path planning technique to apply the proposed procedure to two practical applications. We minimize the time currently required by an industrial vision inspection system to complete a multi-goal cycle, where the neighborhoods are defined using piecewise cubic splines in a seven-dimensional configuration space. Afterwards, we optimize the flight path and the energy consumption of a quadrotor Unmanned Aerial Vehicle (UAV) on an urban survey mission. The specifications of the camera installed on the UAV are used here to define the neighborhoods as three-dimensional polyhedra.

Cum his versare, qui te meliorem facturi sunt.
Illos admitte, quos tu potes facere meliores.
Mutuo ista fiunt, et homines, dum docent, discunt.

Seneca, Epistulae Morales, VII, 8

A Lino

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In this chapter, we first provide a literature review about recent works relevant to our research, and we illustrate their limitations for practical applications. Then, we state our contribution and summarize the obtained numerical results.

1.1 Literature Review

1.1.1 The TSPN

There are optimization instances in which the standard Traveling Salesman Problem (TSP) formulation cannot fully capture the exact nature of the problem. Indeed, if the vertices are allowed to move in certain continuous domains (neighborhoods), not only an optimal Hamiltonian cycle has to be found that visits each vertex once, but also the optimal position of each vertex in its neighborhood has to be defined. The combination of an optimal Hamiltonian cycle and optimal vertices positions is called *optimal tour* hereafter. This problem was initially introduced by Arkin and Hassin [9], and it is commonly referred to as the TSP with Neighborhoods (TSPN).

Some technical problems have been recently posed in the literature where the TSPN formulation seems to be a suitable approach to properly capture their nature. Utility companies employ automated meter reading (AMR) based on radio frequency identification (RFID) to read meters from a certain distance. The reader has thus to plan in advance the shortest path that travels within a certain radius from each meter to minimize the reading costs [47, 93]. The same scenario occurs when mobile robots have to acquire data from distributed sensors and therefore they have to approach each sensor from a minimum distance to allow the wireless communication working properly [105]. Unmanned aerial vehicles (UAV) can be deployed to monitor a set of sites. A flight path has thus to be calculated such that the UAV flies within a certain distance from the center of each site, while minimizing the flying time or the fuel consumption [62]. Industrial manipulators can be used to perform a sequence of multiple tasks during an operating cycle. If the robotic system has 7 or more degrees of freedoms (DOF) there is an infinite number of possible configurations that can be used while performing each task in the sequence. Thus, an optimal sequence of optimal configurations has to be calculated by a multi-goal path planner [44]. In all but the last cited cases the neighborhoods are represented by balls in $\mathbb{R}^2$ or in $\mathbb{R}^3$, and only in the last application the neighborhoods are non-convex regions in the robot configuration space. Depending on the topology of the neighborhoods and on the type of the edge weighting function, specific heuristic approaches are available in the literature.

1.1.2 Heuristics

In the case of partially overlapping or disjoint balls in $\mathbb{R}^2$ and Euclidean norm, indicated also as Close Enough TSP (CETSP), Gulczynski et al. [47] propose different heuristics based on tiling, sweeping circles, radial adjacency, and Steiner zones. Dong et al. [28] propose two heuristics to extract representatives vertices for each neighborhood based on tiling or convex hulls, and then simulated annealing is used to search for a near *optimal tour*. An extension of the Steiner zones heuristic is provided by Mennell [74], where also balls in $\mathbb{R}^3$ and Manhattan norm are considered. First a graph reduction is performed by finding the intersections of the partially overlapping balls (Steiner zones), and representative vertices are chosen for each zone. Then a classical TSP is solved using these vertices, and finally the solution is improved by solving a continuous touring problem. Among the several variants of the main procedure, Mennell [74] proposes also to discretize the Steiner zones into several representative points, and to employ a Genetic Algorithm (GA) to solve the resulting Generalized TSP (GTSP) [94]. The proposed procedures are applied on a set of test instances and results are compared to the one obtained applying different approaches proposed by other authors. Instances from the same test set will be used in this work to benchmark our proposed method.

In the case of partially overlapping or disjoint balls in $\mathbb{R}^2$ and where aircraft dynamics is considered in defining the edge weighting function, Klesh [62] discusses necessary conditions for optimality and proposes two heuristics, based on a "rubberband" approach or on a GTSP model. Since edges are trajectories rather straight lines, not only the position of each vertex has to be considered while searching for a near *optimal tour*, but also the derivative of the trajectory at each vertex location.

In the case of disjoints balls in $\mathbb{R}^2$ and Euclidean or Manhattan norm Yuan et al. [105] propose a two step approach: a permutation of the neighborhoods is found by a traditional TSP algorithm, and then an evolutionary approach is employed to find the best point in each neighborhood.

In the case of multi-goal path planning for redundant robotic systems, where the problem complexity is further increased by the fact that a collision free path between neighborhoods has to be found, Gueta et al. [44] propose to find first a near optimal sequence in three steps: (1) cluster the representative placements in the workspace, (2) solve the resulting TSP in each cluster, and (3) concatenate the resulting paths. Then each neighborhood is sampled, and a configuration is chosen for each neighborhood by combining a greedy nearest neighbor method and the Dijkstra algorithm using a rough-to-smooth procedure. For similar cases, Saha et al. [91] propose first to extract a small number of discrete samples for each neighborhood. The resulting GTSP is then approximated by calculating a minimum group spanning tree as a special case of the Steiner tree problem [85] and by performing a preorder tree walk.

The main limitation of the mentioned heuristic approaches is the fact that only balls in $\mathbb{R}^2$ or in $\mathbb{R}^3$ are employed as neighborhoods. Only in the case of robotic manipulators also non-convex neighborhoods are considered, but a pre-sampling step is performed to transform the TSPN into a GTSP. In this case, to avoid an excessive complexity in the GTSP model, only few samples can be used to replace the continuous neighborhoods with clusters of nodes [91].

1.1.3 Approximation algorithms

Besides heuristic approaches, in the computational geometry literature many approximation algorithms have been proposed for the case of the TSPN in $\mathbb{R}^2$ with Euclidean norm. The achieved approximation factors vary for the different cases depending on the fact that the neighborhoods may be connected or non-connected, disjoint or intersecting, with comparable or varying diameter, convex or non-convex, and fat or non-fat. Two definitions of fatness are available in the literature:

- A region $O \subseteq \mathbb{R}^2$ is said to be α-fat if for any disk D, which does not fully contain O and whose center lies in O as illustrated in Figure 1.1,

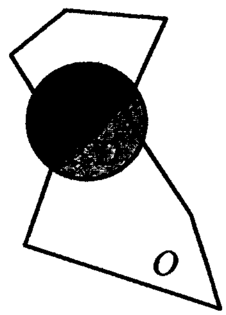

Figure 1.1: A disk D used in the definition of fatness for a region $O \subseteq \mathbb{R}^2$.

the area of the intersection of O and D is al least $1/\alpha$ the area of D [30]. For example α is 1 for plane, 2 for the half plane, 4 for disk, ∞ for a line segment.

- A region O is said to be α-fat if the ratio of the radius of the smallest circumscribing circle to the radius of largest inscribed circle is bounded by α [77].

Non-fatness and intersection seem to make the problem harder. Moreover, it has been proved that the most general case of TSPN is APX-hard [22, 90], even for the simple case where the neighborhoods are line segments of approximately the same length [30]. For connected, disjoint, varying diameter, and α-fat neighborhoods, Elbassioni et al. [30] propose an $O(\alpha)$-approximation algorithm, which can be extended to $O(\alpha/\sqrt{m})$ in $\mathbb{R}^m$. Under the second definition of fatness, Mitchell [77] proposes a polynomial-time approximation scheme (PTAS) for the same problem topology. For connected, intersecting, and comparable diameters neighborhoods a $O(1)$-approximation is proposed by Dumitrescu and Mitchell [29]. Furthermore, if the neighborhoods are convex and α-fat an $O(\alpha^3)$ approximation is given in [30]. Finally, in the case of connected and intersecting neighborhoods with varying diameter, an $O(\log(n))$ approximation is proposed for polygons [43] and for more general neighborhoods [30], where n is the number of neighborhoods. The latter approximation becomes $O(1)$ if the neighborhoods have comparable diameters.

The above mentioned approximation algorithms, which are polynomial time in many cases and represent a useful tool for finding a valid upper bound to the solution, can deal with several types of neighborhoods but non-connected. However, their deterministic nature may cause them to provide a near *optimal tour* with an effective approximation factor yet too large for practical applications. Yuan et al. [105] have shown that for the simple case of disjoint balls in $\mathbb{R}^2$ the solution provided by his evolutionary approach always outperforms the approximation algorithm provided in [30], although it generally requires a larger CPU time. Moreover, approximation algorithms are based on the Euclidean norm and mainly deal with neighborhoods in $\mathbb{R}^2$, except for the extension to $\mathbb{R}^m$ proposed in [30], while in many technical fields different definitions of the objective function in higher dimensional spaces might be required.

1.2 Contribution

A robotic system is said to be *redundant* if, for reaching a given goal location, it can assume several, possibly infinite, *configurations*. For example, a robot manipulator typically interacts with objects by using a device mounted at the end of its arm and called *end-effector*. If the manipulator has seven or more joints, for placing its end-effector at a given position and orientation in the workspace, its various joints can assume infinitely many possible positions, or configurations. Another example is an UAV that has to acquire a picture of a target. This picture can be taken from an infinite number of positions, or configurations, as far as some given specifications are fulfilled, such as image resolution or distortion.

In practical applications, a redundant robotic system might be asked to reach not only one but n goal locations within an operation cycle. Each goal location i can be represented by a set $\mathcal{Q}_i$ of configurations in the collision free configuration space, $\mathcal{Q}_{free}$, of the robotic system. The set $\mathcal{Q}_i$ is the *neighborhood* for goal i. Given two goals $i \neq j$ and two configurations $\boldsymbol{q}_i \in \mathcal{Q}_i$ and $\boldsymbol{q}_j \in \mathcal{Q}_j$, a cost function for the manipulator to move from $\boldsymbol{q}_i$ to $\boldsymbol{q}_j$ is defined. This function is called hereafter *edge weighting function*, and it is indicated as $\mathrm{d}(\boldsymbol{q}_i, \boldsymbol{q}_j)$. In this work we aim to find configurations $\boldsymbol{q}_i \in \mathcal{Q}_i$ for $i = 1, \ldots, n$ and a tour that connects these n configurations such that its total cycle cost is minimized.

This problem is very complex in its full generality, as neighborhoods can have arbitrary shapes determined by the system specifications or physical constraints. Moreover, computing the optimal path and thus the cost to move between two goal locations is in itself a difficult problem since it involves robot kinematics and obstacle avoidance. In this work we study first a simplified version of this problem using analytically defined cost functions. However, we address most of the limitations of previous approaches illustrated in Section 1.1, allowing wider classes both of neighborhood topologies

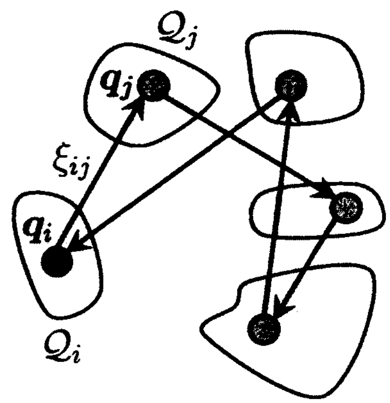

Figure 1.2: An ATSPN instance. The five areas around the vertices shaded in bright blue are the neighborhoods. The directed tour depicted with black arrows is a feasible solution.

and of edge weighting functions. In particular, polyhedra, ellipsoids, and cubic splines in $\mathbb{R}^m$ have been employed as neighborhood, and four types of edge weighting functions have been considered: Manhattan, Euclidean, Quadratic, and Maximum norm. These provide enough flexibility to reasonably estimate the actual system performance for practical purposes. Under these initial assumptions, we propose three novel approaches to search for an *optimal tour*.

The first approach to solve such instances of TSPN is to formulate it as a non-convex Mixed Integer Non Linear Programming (MINLP) using as variables the coordinates of the vertices $\boldsymbol{q}_i$ for $i = 1, \ldots, n$ as well as binary variables ξ_{ij} for $i, j = 1, \ldots, n$ to represent the possible edges of the tour, as shown in Figure 1.2. The resulting MINLP is non-convex, even when the integrality constraints on the variables ξ_{ij} are relaxed. It follows that only solvers for non-convex MINLP problems can be used for its solution, such as BARON [92], COUENNE [12, 20], and LINDOGLOBAL [68].

On the one hand, these solvers struggle to solve relatively small size instances of TSPN. On the other hand, by using a specific feature of the MINLP formulation and customizing the solver by adding specific cut generators and heuristics, we are able to solve instances with up to 16 polyhedral or ellipsoidal neighborhoods far more efficiently. The crucial feature that we exploit is that once all the binary variables in the formulation are fixed to 0 or 1 values, the continuous relaxation of the remaining problem is convex. It is thus possible to solve it to optimality using a continuous solver. For example, the solver COUENNE (with default settings) requires 733 seconds to solve a TSPN instance with ellipsoids in $\mathbb{R}^2$ (tspn2DP6_2) to optimality, while the proposed approach solves it in a fraction of a second.

In the second approach, the problem is reformulated as a convex MINLP instance under the assumption that the neighborhoods and the edge weighting functions are both convex, and three different formulations are derived. Then, using a convex MINLP optimizer such as BONMIN [17, 20] or MOSEK [4] the problem can be solved to optimality. The best performance is obtained when BONMIN is customized to solve TSPN instances, and a specific cut generator is implemented to efficiently handle specific constraints existing in the MINLP formulation of the problem. Using this exact procedure CPU time is improved up to two orders of magnitude, and instances with up to 20 neighborhoods have been solved to optimality.

A third approach is then proposed to handle larger scale TSPN instances with possibly non-convex neighborhoods. Starting from the MINLP framework used in the previous approaches an hybrid random-key Genetic Algorithm (GA) is specifically developed to search for a near *optimal tour*. The choice of a random-key coding for the GA guarantees feasibility during crossover operations, and avoids to explicitly formulate the subtour elimination constraints of the original MINLP formulation resulting in a more efficient representation of the problem. Moreover, the CPU time of the GA is drastically reduced by replacing commonly used mutation operators with two ad-hoc heuristics: (1) the position of each vertex is fixed, and their sequence is improved by using the Lin-Kernighan heuristic [67]; and (2) the position of each vertex is optimized by solving the Non Linear Programming (NLP) instance resulting from fixing their sequence in the original MINLP formulation.

To evaluate the performance of the proposed GA, TSPN instances were either randomly generated or selected among the CETSP problems proposed by Mennell [74]. In the first case, the GA was able to find the *optimal tour* in all the cases where the solution was also calculated using the MINLP optimizers, while improving the computational performance by orders of magnitude. In the second case, the GA improved the best known near *optimal tour* on average by 1.92%, although the proposed approach is not specifically tailored to solve CETSP instances. Finally, it is worth mentioning that a drawback of the proposed heuristic approach is that no approximation factor for the results can be guaranteed. However, in case an upper bound is required, it is sufficient to first run an approximation algorithm if

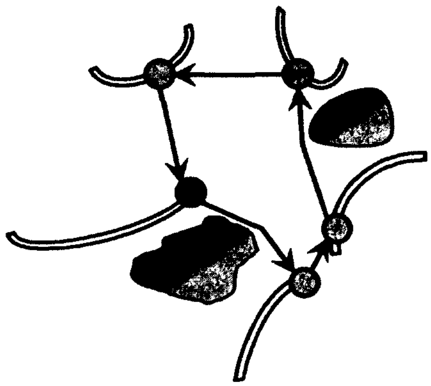

Figure 1.3: Collision-free near *optimal tour* for a TSPN instance. Obstacles are shaded in red/yellow and neighborhoods in bright blue.

available, and then introduce the obtained approximation as a chromosome of the initial population used in the GA.

Finally, to account for collision avoidance and thus to achieve a more realistic evaluation of the edge weighting function for practical applications we embed in the GA a probabilistic path planning technique based on bidirectional Rapidly-exploring Random Trees (RRTs). Figure 1.3 illustrates a TSPN instance where obstacles are considered in the definition of the near *optimal tour.* Moreover, we integrate a dynamic simulator within the GA to optimize the energy consumption of the considered robotic system.

In particular we apply the proposed approach to two test cases. First, we minimize the cycle time of a 7 DOF robotic vision inspection system. The neighborhoods are here approximated using piecewise cubic splines in a seven-dimensional configuration space, and the employed edge weighting function is based on the Quadratic or the Maximum norm. For the specific scenario considered in this work with a 32-goal cycle, experimental tests show an improvement of the current cycle time up to 30%. Second, the flight path and the energy consumption of a quadrotor drone on an urban inspection mission are optimized. The neighborhoods are here defined as three-dimensional polyhedra and the edge weighting function is either the Euclidean or the Quadratic norm. The level of the archived improvement with respect to the results obtained with more traditional optimization techniques generally depends on the number of the neighborhood and their spatial distribution. Within this work we observe the best improvement on the largest analyzed instance with more than 1,500 goals, where path length and energy consumption are improved up to 38% and 23%, respectively.

The thesis is organized as follows. The used MINLP formulation is presented in chapter 2 together with the first two solution procedures. The hybrid random-key GA is presented in chapter 3. The two considered robotic applications are illustrated in chapters 4 and 5. Finally, chapter 6 contains conclusions and discusses potential future work.