Multi-Goal Planning for an Autonomous Blasthole Drill

Pantelis Elinas
Australian Centre for Field Robotics (ACFR)
The University of Sydney, Australia, NSW 2006
p.elinas@acfr.usyd.edu.au

Abstract

This paper presents multi-goal planning for an autonomous blasthole drill used in open pit mining operations. Given a blasthole pattern to be drilled and constraints on the vehicle's motion and orientation when drilling, we wish to compute the best order in which to drill the given pattern. Blasthole pattern drilling is an asymmetric Traveling Salesman Problem with precedence constraints specifying that some holes must be drilled before others. We wish to find the minimum cost tour according to criteria that minimize the distance travelled satisfying the precedence and vehicle motion constraints. We present an iterative method for solving the blasthole sequencing problem using the combination of a Genetic Algorithm and motion planning simulations that we use to determine the true cost of travel between any two holes.

Introduction

In this work, we introduce the problem of blasthole pattern drilling in open pit mining and propose an iterative method for solving it providing early results supporting our method. Given a drilling pattern, an autonomous drill must decide on the order in which to drill the holes taking into account its operational constraints along with the drilling guidelines and safety rules decided by the mining company.

Blasthole pattern drilling is a Sequential Ordering Problem (SOP). SOPs are Asymmetric Travelling Salesman Problems (ATSPs) (Applegate et al. 2006) with the addition of precedence constraints, i.e., some cities must be visited before others for all tours that are valid solutions of the SOP. What makes our problem difficult to solve are dynamic obstacles, e.g., already drilled holes, vehicle motion constraints, large drilling patterns, and the orientation of the drill which must be taken into account when deciding on the cost to travel from one hole location to the next. The latter generalizes the hole sequencing problem to a group Steiner problem which is notoriously difficult to solve even though a quasi-polynomial approximation technique was recently published (Even and Kortsarz 2002).

In order to address the above issues, we propose the use of an iterative, hybrid method combining an SOP solver and a motion planning/simulation module to generate good plans for large drilling patterns. Our overall approach most closely resembles the work of (Saha et al. 2006) on multi-goal planning for industrial robots. Our method differs in (a) how we solve the SOP, (b) the planning method for a non-holonomic vehicle, (c) determination of the vehicle's orientation, and, (d) the overall application to surface mining.

Because of the combinatorial complexity of solving an SOP, we make use of an approximate optimization algorithm. Genetic Algorithms (GAs) (Chen and Smith 1996) have been shown to provide good approximate solutions for large SOPs and so we utilize one such method in this work. An alternative approach is a recently proposed Ant Colony Optimization algorithm (Gambardella and Dorigo 2000) which for some problems can provide solutions with tighter bounds around the optimal tour.

In order to recover the true cost of travelling from one hole location to the next, we use a randomized, motion planning algorithm namely a Probabilistic Roadmap (PRM) (Kavraki et al. 1996). The motion planner generates a valid path connecting two configurations of our vehicle modeled using non-holonomic constraints. We connect states using curvature polynomial steering functions (Kelly and Nagy 2003) that can be computed efficiently, are easy to execute, and can be used to optimize any desirable performance index such as trajectory smoothness or fuel consumption.

Problem definition and previous work

The drill operator is given a pattern of holes that must be drilled at a particular location/bench in the pit. This pattern includes the location (specified using GPS coordinates) of the required holes and their depth along with other information such as the type of drill operation that should be used (i.e., rotary or percussion) for each hole. Other information included is not relevant to the sequencing problem. Operations follow in a tram-jack-drill-jack cycle.

This paper addresses the problem of finding the order in which to drill the holes in a given blasthole pattern. This sequence is not dictated *a priori* and it is currently determined by the drill operator based on a set of general guidelines and priorities for various classes of holes. Briefly, these rules dictate that the first holes to be drilled are "angle holes" along the open face, which constrains the drill's orientation so as to be aligned perpendicular to this edge. These are

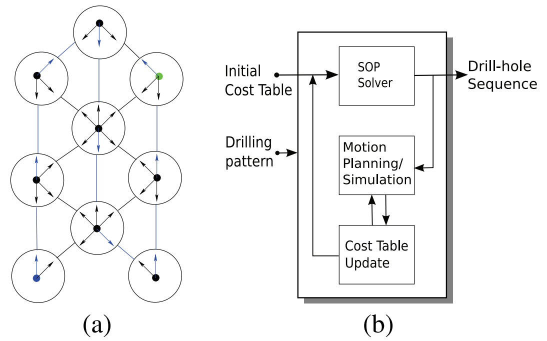


Figure 1: (a) Blasthole sequencing shown as a group Steiner problem with each vertex in the partitioned graph representing the drill at the same spatial location but having different orientations. (b) A block diagram of our proposed solution method. Given an initial cost table, we employ an optimization algorithm to solve the sequential ordering problem. Given this solution, we perform simulation to re-estimate the travel costs between holes. This iterative procedure ends when convergence is reached or a stopping criterion is met.

followed by holes near free face(s) or adjacent to blasted material (i.e., a high wall or a crest). After these are batter and buffer holes, which are followed by drilling of the main rows. The main rows are aligned in straight lines making it straightforward for the operator to drill.

The drill is a tracked vehicle capable of turning on the spot towards any direction. However, its massive size (15 meters long and 12 meters wide with a max speed of 1m/s) and potential to damage the drilling surface makes accurate positioning over target hole locations difficult. For the purposes of this paper, we model it as a non-holonomic car-like vehicle with a constrained turning radius. This model allows us to generate smooth motion plans with constrained curvature which when executed should minimize surface damage. Moreover, turning on the spot is not advisable since such motions increase the chances of damaging an already drilled hole by pushing dirt back down the hole requiring drilling it a second time, a very time consuming and undesirable action. Finally, the operator selects the first and last hole to drill such that the above requirements are satisfied. The last hole is chosen in a location that allows the drill easy access away from the bench. In our work, we assume that the first and last holes are given along with any precedence constraints as described earlier.

As given above, blasthole pattern drilling is a combinatorial optimization problem which in the most general case is similar to a group Steiner problem with precedence constraints. Part (a) of Figure 1 presents the drill pattern drilling problem as a group Steiner problem. The Figure shows a pattern with 9 holes arranged in 3 columns. A small set of drill orientation are shown using arrows. The solution to the sequencing problem is to select a tour through each of the partitioned nodes selecting one orientation out of the sampled set such that some optimization criterion is minimized. A possible solution is illustrated using the blue color while the first and last holes have blue and green colors respectively.

Planning over a partitioned graph of goals is hard and at best there exist only approximate quasi-polynomial solution methods that can guarantee the approximation is bounded (Chekuri, Even, and Kortsarz 2006); this latter method has been used to solve a multi-goal planning problem for an industrial robot arm (Saha et al. 2006). We propose to use a hybrid, iterative solution algorithm solving our drilling problem posed as a Sequential Ordering Problem coupled with a probabilistic motion planning method for recovering the true cost of navigating from one hole to the next.

Other researchers have solved similar multi-goal path planning problems in different domains. For example, (Spitz and Requicha 2000) consider the path planning problem for coordinate measuring machines. Moreover, (Danner and Kavraki 2000) consider a similar approach for solving the "watchman problem" in robotics. Lastly, (Saha et al. 2006) solve another multi-goal path planning problem for an industrial, spot-welding robotic arm. Our method most closely matches theirs with the differences being that (a) we solve an SOP instead of a TSP; (b) our robot is a non-holonomic vehicle instead of a robotic arm; and (c) our solution method reduces the partitioned graph problem to one with singleton groups by heuristically selecting a preferred individual, i.e., orientation, in each sub-group.

Proposed Solution Method

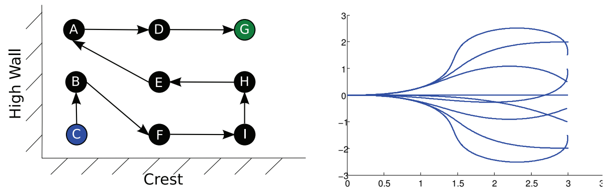
We have already established that the blasthole drill planning problem is hard. In this section, we propose an iterative method for solving this problem first by solving the SOP using an initial cost table, then performing a simulation of the vehicle drilling the pattern in the computed order, and finally recalculating the SOP cost table; this procedure is to be repeated until convergence is achieved or some other termination criterion such as maximum number of iterations is met. Convergence is achieved if the tour with minimum cost remains the same or within a predefined threshold for two consecutive iterations. Part (b) of Figure 1 shows a block diagram of our proposed iterative procedure. In the following 3 Sections, we describe the components of our proposed method and the drill orientation selection mechanism.

SOP solver

We solve the Sequential Ordering Problem using the commonality-based Maximum Partial Order/Arbitrary Insertion (MPO/AI) genetic algorithm of (Chen and Smith 1996).

Part (a) of Figure 2 shows a graphical representation of a small drilling example with 9 holes positioned on a regular grid. For this example, we do not consider the vehicle orientation at each location but only its position in $2D$. The starting location C is shown in blue and the desired last location G is shown in green. The precedence constraints for this problem are defined as follows:

1. Crest (angle-holes): C before B, F before E, and I before H



(a) (b)

Figure 2: (a) An example solution for a drilling pattern with 9 holes. The first hole is shown in blue and the last hole is shown in green. The arrows denote the best solution found using a Genetic Algorithm with the MPO/AI crossover operator satisfying our specified precedence constraints (see text). (b) Examples of using cubic polynomial steering functions to connect two arbitrary states. In all cases the start pose is $\{0, 0, 0, 0\}$; moreover, the curvature for all goal poses is set to 0. In all cases the final pose is correct.

2. High-wall (face/adjacent holes): A before D, B before E, and C before F

These constraints satisfy the established drilling guidelines that holes near a crest or a high wall must be drilled first. For this experiment, the costs to travel between any two holes were set to the Euclidean distance between them. In this particular case, the cost table is symmetric. The solution is also shown in Part (a) of Figure 2 using arrows. Inspecting the solution, we see that all the constraints are satisfied.

Motion Planning

We perform motion planning to compute the cost of travel from one hole location to the next. We have obtained good results using curvature polynomial steering functions (Kelly and Nagy 2003) in conjunction with a Probabilistic Roadmap (PRM) framework. We have to perform many queries at each iteration of our algorithm and the PRM method is much better suited for our application since the roadmap can be constructed once at the beginning and then used to answer all queries. The polynomial steering functions are also fast to compute and can be designed to optimize for any performance index that is suitable to our application, e.g., trajectory smoothness, fuel consumption, or GPS signal strength. These primitives are also much more straightforward to execute by a controller. Part (b) of Figure 2 shows a small set of curvature polynomials connecting the same start state to arbitrary goal states.

For the PRM, we transform all queries from the global coordinate frame to a local coordinate frame with respect to the vehicle's body frame at the start state; in other words, the start pose is always at the origin of the local coordinate frame and has 0 orientation and curvature. Further, we constrain our roadmap to a small region around the starting state setting the bounds on position to be in the range -25 to 25 meters. Orientation can be anywhere in the range 0 to 2π. We have obtained good results using a PRM with 2000 states sampled uniformly at random.

Given the roadmap and a query, we search for a path that connects the start and goal states. We use the A* algorithm to efficiently search for a path. The cost to travel from one vertex to another, i.e., an edge in the graph, is given by the length of the curvature polynomial. We specify the heuristic function as the Euclidean distance between an intermediate state and the goal.

Orientation Selection

Computing the optimal sequence is mostly complicated by the fact that with the exception of special holes the rest can be drilled with the vehicle at any orientation. We explained earlier how this entails blasthole drill planning to be a group Steiner problem for which currently do not exist solution algorithms that can efficiently solve problems as big as ours, i.e., in most cases, well over 100 holes and arbitrary orientation at each location.

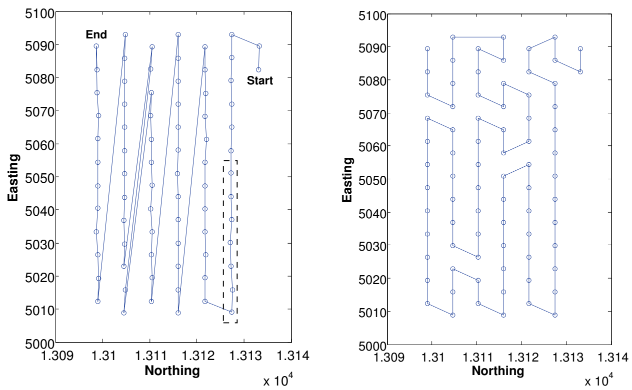
As a result, we propose to simplify the problem to the easier to solve case of a non-partitioned graph considering a single configuration at each hole location. There is no way to decide what the correct orientation is without performing an exhaustive search which is computationally infeasible. So, we select the vehicle's orientation employing a heuristic based on our intuition.

- **The two-step lookahead heuristic:** We consider setting the orientation according to the direction of motion two-steps ahead. The order is the one selected in the last tour computed given the most recently updated cost table. To give an example of setting the vehicle's orientation using this method consider the simple pattern shown in Part (a) of Figure 2 and assume that the vehicle is at location C facing North towards location B. The orientation at B after completing the motion from C to B will be in the direction of location F which is the next in the tour.

Empirical Evaluation

Here we present an early result of evaluating our method by comparing the computed plan to an operator's with data from a surface mining operation. Part (a) of Figure 3 shows the pattern as drilled by the operator. He started by drilling the rightmost column where the angle holes are located (shown with a dashed rectangle at the bottom right) and then proceeded to drill the columns one at a time. When drilling the bottom two holes in the second column from the left and top two holes from the third column from the left, he performed a maneuver that is uncommon; we do not know why he did this. Regardless, we consider his solution as the ground truth and compare it with that of our proposed method.

Part (b) of the same Figure shows the best tour selected by our algorithm after 25 iterations at which point convergence was reached with a tour length of just below 825 meters. Figure 4 shows the moving average for the tour length at the end of each iteration. We can see that it reaches a minimum after about 25 iterations. For our experiment we used a population of 500 individuals for the genetic algorithm and initialized the cost table to twice the Euclidean distance between every pair of holes. This result shows that our method is capable of discovering that large parts of the pattern are designed to be drilled with the vehicle moving in



(a) (b)

Figure 3: (a) The original pattern (77 holes) and how it was drilled by the operator. (b) The result obtained using our iterative method after 25 iterations.

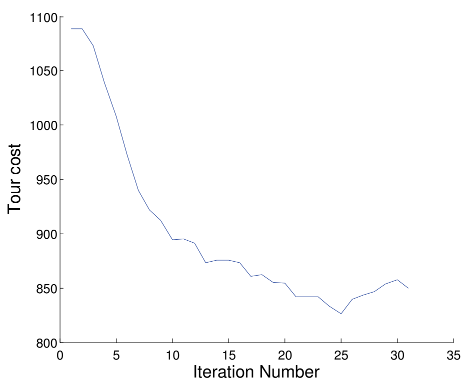


Figure 4: Plot of the moving average of the tour length (in meters) versus iteration number solving the pattern shown in Figure 3.

straight lines. The precedence constraints for the crest holes were also satisfied in the final solution. The solution is close to the operator's which provides some promising evidence of the usefulness of our method. Convergence was reached in 27 minutes including the time to construct the roadmap (about 5 minutes.)

Conclusions and Future Work

In this paper, we introduced the blasthole sequencing problem in surface mining and proposed an iterative, approximate method for solving it. Given a pattern to drill, precedence and orientation constraints for some of the hole locations, and constraints on the vehicle's motion, we showed how the sequencing problem can be posed as an SOP and solved using an iterative procedure such that the true cost to travel between any two hole locations for given start and goal orientations is estimated using a motion planning stage. We also presented a heuristic for selecting the vehicle orientation at each hole location and an initial evaluation of our method with data from real operations.

In the future, we plan to further evaluate our method using a larger data set. Also, we wish to integrate the motion planner with the construction component of the genetic algorithm. We believe that the latter will be required for finding solutions for more complex patterns and also for online updating of the solution. External factors such as those that caused the operator to perform an uncommon action in the example shown earlier cannot all be accounted for in simulation and as such an online plan refinement method is necessary.

Acknowledgements

The author would like to thank Florian Oppolzer and Ross Hennessy for providing the drilling data for the experiments and Surya P. N. Singh for valuable discussions in formulating the ideas presented here. This work is supported by the Rio Tinto Centre for Mine Automation and the ARC Centre of Excellence programme funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

Applegate, D. L.; Bixby, R. E.; Chvatal, V.; and Cook, W. J. 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.

Chekuri, C.; Even, G.; and Kortsarz, G. 2006. A greedy approximation algorithm for the group Steiner problem. *Discrete Appl. Math.* 154(1):15–34.

Chen, S., and Smith, S. 1996. Commonality and genetic algorithms. Technical Report CMU-RI-TR-96-27, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Danner, T., and Kavraki, L. 2000. Randomized planning for short inspection paths. volume 2, 971–976. San Fransisco, CA: IEEE Press.

Even, G., and Kortsarz, G. 2002. An approximation algorithm for the group Steiner problem. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, 49–58. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Gambardella, L. M., and Dorigo, M. 2000. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS J. on Computing* 12(3):237–255.

Kavraki, L. E.; Svestka, P.; Latombe, J.-C.; and Overmars, M. 1996. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4):566–580.

Kelly, A., and Nagy, B. 2003. Reactive nonholonomic trajectory generation via parametric optimal control. *The Int. Journal of Robotics Research* 22(7 - 8):583 – 601.

Saha, M.; Roughgarden, T.; Latombe, J.-C.; and Snchez-Ante, G. 2006. Planning tours of robotic arms among partitioned goals. *The International Journal of Robotics Research* 25(3):207–223.

Spitz, S. N., and Requicha, A. A. G. 2000. Multiple-goals path planning for coordinate measuring machines. In *Proc. IEEE Int. Conf. On Robotics and Automation*, 2322–2327.