Planning Dragline Positioning Sequence with A* Search Algorithm ⋆

Haoquan Liu * **Michael Kearney** * **Michael Forbes** **

* *School of Mechanical and Mining Engineering, The University of Queensland, Brisbane, QLD 4072, Australia (e-mail: h.liu@uq.edu.au, m.kearney@uq.edu.au).*
** *School of Mathematics and Physics, The University of Queensland, Brisbane, QLD 4072, Australia (e-mail: m.forbes@uq.edu.au)*

Abstract: Dragline excavators are among the largest earthmoving machines in surface mining, where they are used to remove the waste material that sits above a target mineral or coal reserve. The effectiveness of their operation is highly dependent on the sequence of positions at which they operate and the material movement during the excavation at each position. This paper addresses the problem of how to position a dragline in order to efficiently finish an excavation task with a specified digging and dumping strategy at each position. The dragline positioning sequence is computed among a range of candidate positions using the A* search algorithm. The algorithm is applied to two simulated dragline excavation scenarios that are defined based on actual terrain data. We show that the proposed A* algorithm is able to produce the optimal positioning sequences in a reasonable amount of computation time, allowing it to be applied in real-time decision support for the dragline operators.

Keywords: Artificial intelligence, process optimization, dragline operation planning, decision making.

1. INTRODUCTION

Figure 1 shows a typical dragline along with its major mechanical components. Draglines are extensively employed to excavate a long strip of waste material (also called overburden) to expose a seam of target mineral or coal. Each dragline is a self-contained excavation system that achieves the movement of waste material from one designated area to another without the aid of other equipment. Figure 2 is a graphical representation of an actual coal mining strip for the dragline. A strip of overburden is gradually removed while the dragline moves through a sequence of positions above the overburden and performs cyclic digging and dumping operations at each position. Specifically, each dig-dump cycle comprises: filling the bucket by dragging it towards the fairlead with the drag rope; hoisting the loaded bucket with the hoist rope while swinging it to a chosen location over the adjacent dumping area; dumping the material; and swinging and lowering the bucket back to the digging area to start the next cycle. A spoil pile is formed and extended in the adjacent area by the dumped material. The drag rope and the hoist rope are controlled by separate motors that are located inside the machinery house. A dragline repeats this dig-dump cycle approximately one thousand times a day with a cycle time of around one minute (Humphrey and Wagner, 2011; Mirabediny, 1998).

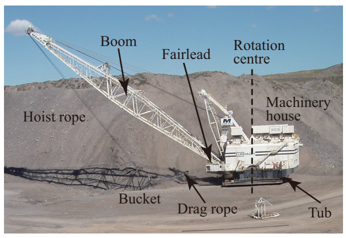
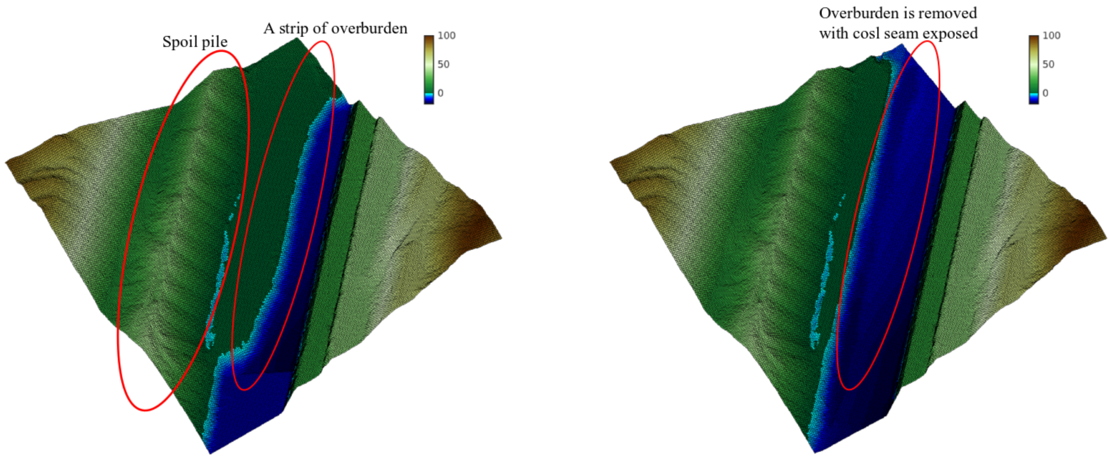


Fig. 1. A typical dragline (from Curragh Queensland Mining Limited (2012) with annotations). The position of a dragline refers to the location of the rotation centre around which the dragline can swing.

Improvements of the dragline excavation efficiency can bring significant economic benefits. According to Corke et al. (2006), a 1% improvement in dragline productivity is valued at around AUD $1M per machine per year. The dragline productivity, which is generally measured as the rate of material removal, is significantly affected by the operators' decisions about: (i) the sequence of positions at which the dragline operates (positioning); (ii) the material to be removed at each dragline position (digging); and (iii) the placement of the dug material at each dragline position (dumping). Currently these decisions are made by the operators, and the excavation is largely dependent on their skill and experience. Optimizing the excavation strategy helps them achieve consistent and effective dragline operations. Therefore, the operational variance in the excavation is reduced and the productivity is improved. These two benefits are consistent with the aims of automating the dragline excavation.

⋆ The first author would like to appreciate the support of the Australian Government Research Training Program Scholarship during his research.



(a) Current terrain. (b) Target terrain.

Fig. 2. An actual coal mining strip for the dragline. The left figure shows the current terrain of the strip with the spoil pile that is formed after previous dragline operations. The right figure shows the designed target terrain after the removal of the whole strip. The average depth of the overburden is 16 metres.

The automation of a dragline excavating the overburden within a designated area can be broken down into three levels of tasks. At the top (strategic) level, the dragline positioning sequence is determined, along with the patches of material to be removed and the locations at which the material is dumped. The planner must ensure the feasibility of this strategy, meaning that the excavation task can be finished by the dragline following this strategy. Based on the strategy from the top level, the planner at the second (tactical) level determines the digging and dumping actions for each dig-dump cycle, including the starting and ending points of filling the bucket and the location to release the full bucket over the spoil pile. At the third (execution) level, the planner (or controller) determines the inputs to the actuators (electric motors) so that the bucket automatically follows desired trajectories to execute the determined digging and dumping actions at each cycle. The hierarchical nature of these levels is that the decisions are imposed downwards, while the geometric and operational constraints are imposed upwards.

Most existing works towards the automation of dragline excavation have focused on the third level of the task (Corke et al., 2006, 1997; Winstanley et al., 1997, 2007). The dynamics of the dragline system was modelled and control methods were applied to automate the process of swinging the bucket between a predefined digging point and dumping point. Our work concerns planning for the top level, where little work has been reported in the open literature. Sier (1993) and Mirabediny (1998) approached this problem by modelling the digging and dumping process as relocating sub-blocks of material to the destination locations above the existing spoil pile for the given the dragline positions. However, they only considered the operation over 2D sections of the mining strip. The project conducted by Thornton and Whiten (2003) applied genetic algorithms to find 'good' excavation strategies for draglines in both 2D and 3D scenarios. However, their work did not capture the actual dragline operational constraints and therefore, the resulting strategies are practically infeasible to implement.

In this paper, we address the problem of planning the dragline positioning sequence in order to finish the excavation task of a designated area in a minimum amount of time using a specified digging and dumping strategy. The positioning sequence is major factor in dragline excavation as it determines the digging region that the dragline can cover and the feasible dumping locations that the dragline can utilize. Therefore, the use of a specified digging and dumping strategy for each position is a sensible way to reduce the complexity of the planning problem. Solving this problem helps us better understand the desired walking pattern of the dragline excavator and contributes to the development of more general dragline operation planning framework in the future.

The dragline positioning problem is considered as a sequential decision making problem. The solution of this problem is a sequence of actions (where to move) that transform the state of the system (terrain profile) from its initial state to a goal state with a minimum total cost. We construct a tree to represent this problem, where each node represents a state and each edge represents the action initiating the transition from one state to another state. We use the A* search algorithm to compute a path from the root to a goal node in the tree (Hart et al., 1968). Initialized with the root, a set of unexpanded nodes (*OpenSet*) is iteratively updated by selecting a member to further explore, adding all the children of the selected node to *OpenSet*, and then deleting the selected node from *OpenSet*. The algorithm terminates when the selected node is a goal node. From *OpenSet*, the A* algorithm chooses to expand the node through which the path from the root to the goal has a minimum overall cost. The overall cost of a path through a node is the sum of the actual cost of the path from the root to this node and the estimated cost of the path from this node to the goal (heuristic cost). It guarantees to produce the optimal solution after termination if the heuristic cost is admissible, that is it never overestimates the cost from the current node to the nearest goal node. The A* algorithm had extensive application in planning problems (Hansen and Zilberstein, 2001; Koenig et al., 2004; Szer et al., 2005).

The remainder of this paper is structured as follows. Section 2 formally describes the dragline positioning problem. Details of the A* algorithm applied for this problem are shown in Section 3 . Section 4 shows the simulation results from applying the algorithm to two dragline excavation scenarios. Section 5 concludes the findings in this paper and proposes future work.

2. PROBLEM FORMULATION

The terrain profile in the dragline working environment is modelled using uniform height grids, where each grid point is represented by its horizontal x-y coordinates and its vertical height z (Li et al., 2005). Three areas are designated in the environment: i) the digging area; ii) the dumping area; and iii) the working area. The dragline moves within the working area and when it is fixed at a position, it digs the overburden in the digging area and dumps the dug material directly onto the spoil pile that lies in the dumping area. A target terrain profile is defined for the digging area based on the height level of the coal seam. A set of grid points underneath which the coal is buried is denoted by $\mathcal{G}_c$. The excavation task is defined as removing the overburden within the digging area until a specified number of grid points in $\mathcal{G}_c$ reach their target heights. Our goal is to obtain the positioning sequence following which the dragline is able to finish the excavation task with a minimum amount of time using a specified digging and dumping strategy.

3. A* ALGORITHM FOR DRAGLINE POSITIONING

3.1 Search Tree Structure

In this problem, each node in the search tree corresponds to the terrain profile of the digging and dumping areas. Since the digging and dumping strategy is fixed for all dragline positions, the transition from the current state to the next state is determined by the selection of the next dragline position. Therefore, the action space at each state is the set of candidate positions that the dragline can move to next. The action sequence from the start node to a certain node is unique and irreversible due to the geometric constraints in the excavation. Each node (except the root) has a unique parent in the tree. An example search tree for this problem is shown in Figure 3.

3.2 State Transition

After an action (the new dragline position) is selected at the current state during searching, the next state is determined by applying the specified digging and dumping strategy to the current terrain at the new position. The chosen strategy must capture the effect of the material movement on the terrain and the constraints imposed by the machine on what can be dug and where the material can be dumped. This ensures that the resulting excavation strategy is practically feasible to implement at the tactical and execution planning levels.

For the digging and dumping strategy applied in this work, the dragline removes all the material it can reach

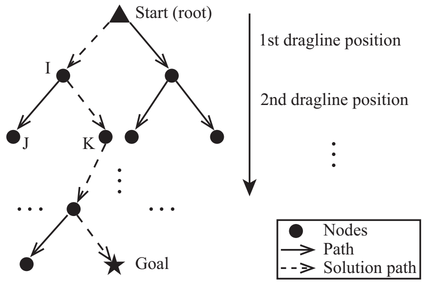


Fig. 3. A simple example of the directed rooted tree constructed in this problem where there are only 2 candidate dragline positions. Here Node I is the parent of Node J and K.

in the digging area at each position and dumps it to the nearest dumping location, subject to the available spoil room. This strategy is described in Algorithm 1. A sequence of dumping locations are defined over the spoil pile for each dragline position, as shown in Figure 4. A specific swing angle is computed for each dumping location, which is used in determining the swing cost. For the dumping strategy specified, the dragline always dumps the material to the location with the smallest swing angle until its dumping capacity is reached or there is no more available material in the digging area. Given the current terrain profile and dragline position, the maximum volume of material that can be dumped ($V_{p,max}$) and the corresponding dumping operation $a_{p,max}$ are first computed by **MaxDump** under this strategy. The maximum volume of material that can be removed from the digging area ($V_{g,max}$) and the corresponding digging operation $a_{g,max}$ are then computed using **MaxDig** by maximizing the digging angle for each grid point. The digging angle for a specific grid point is illustrated in Figure 5. If all the material removed from the digging area can be fit in the dumping area ($V_{g,max} \leq V_{p,max}$), then the final digging operation a_g is assigned to be $a_{g,max}$ and the final dumping operation a_p is calculated by **DumpVolume**; otherwise, a_p is assigned to be $a_{p,max}$ and a_g is calculated by **DigVolume**. The bisection method is used in **DumpVolume** and **DigVolume** to guarantee conservation of material between digging and dumping. In **DumpVolume**, it is used to adjust the volume of the material dumped at final dumping location until the dumping volume equals $V_{g,max}$. In **DigVolume**, the bisection method is used to adjust the digging angles of the grid points in the digging area uniformly until the digging volume equals $V_{p,max}$. Details of the geometric constraints in dragline digging and dumping can be referred to in Liu et al. (2016).

3.3 Cost Functions

Two cost functions are required for the A* algorithm. The first cost function computes the time required to perform digging and dumping with the specified strategy at a new dragline position. In this work, this one-step state transition cost is calculated by

$$\mathbf{Cost}(Current, Child) = t_{prep} + t_{move} + t_{sw}, \qquad (1)$$

where t_{prep} represents the time to prepare the dragline to move to another position, t_{move} indicates the time in moving the dragline to another position and t_{sw} considers

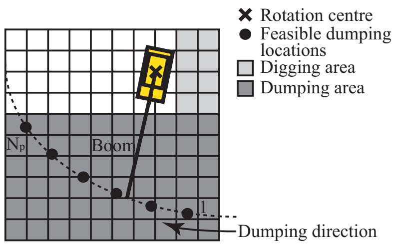


Fig. 4. A schematic showing the feasible dumping locations at a dragline position. The dragline is assumed to perform dumping from Location 1 to N_p until all the available overburden is removed or the maximum dumping capacity is reached.

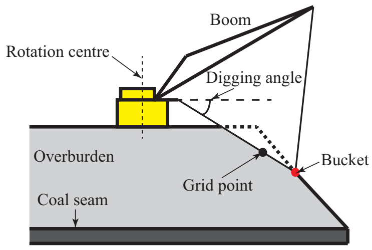


Fig. 5. A schematic showing a sectional view of the digging angle for the particular grid point in the digging area. The dragline bucket is assumed to be a point mass.

Algorithm 1 DigDumpStrategy

Input State $\mathcal{S}$, dragline position Pos
Output Digging and dumping operations a_g, a_p

```
[a_{p,max}, V_{p,max}] ← MaxDump(S, Pos);
[a_{g,max}, V_{g,max}] ← MaxDig(S, Pos);
if V_{p,max} ≥ V_{g,max} then
    a_g = a_{g,max};
    a_p = DumpVolume(S, Pos, V_{g,max});
else
    a_p = a_{p,max};
    a_g = DigVolume(S, Pos, V_{p,max});
end if
return a_g, a_p;
```

the time spent in swinging the bucket between the digging area and the dumping area during excavation. t_{prep} is a fixed cost in this problem. t_{move} is computed given the moving speed of the dragline and the distance between two dragline positions. t_{sw} is calculated based on the aggregated swing angle during digging and dumping. As the material movement is considered in bulk, the computation of the aggregated swing angle is divided into two parts, using a reference boom position shown in Figure 6. The first part considers the total swing angle from the dug grid points in the digging area to the reference boom position and the second part considers the total swing angle from this reference boom position to the dumping locations where the material is dumped. The choice of this position is independent of the calculation.

The other cost function required in the algorithm is the heuristic cost function that is used to estimate the future operation time to finish the task from the current state. In this paper, the heuristic cost for the given state S is computed by

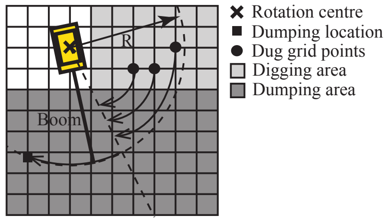


Fig. 6. Calculation of the total swing angle. The dashed line through the rotation centre indicates the reference boom position chosen to divide the calculation into two parts..

$$\textbf{Heuristic}(S, Goal) = t_{h,sw}\frac{V_{min}}{V_{ca}} + t_{prep}, \tag{2}$$

where V_{min} and V_{ca} are the minimum volume of material required to be removed in order to reach the goal state and the dragline bucket capacity, respectively. $t_{h,sw}$ is a constant that estimates the swing time in a dig-dump cycle. To compute V_{min}, the grid points in $\mathcal{G}_c$ are sorted in the ascending order of the differences between their current heights and their target heights. Then, the minimum number of grid points in $\mathcal{G}_c$ needed to reach their target heights (N_{min}) can be calculated given the goal condition. V_{min} is computed by adding the volume of the remaining material at the first N_{min} grid points in $\mathcal{G}_c$. The preparation time for the dragline to reposition is also included in the heuristic cost function as the dragline has to move at least once in order to reach the goal state. The formulation of the heuristic cost ensures that it is admissible. Therefore, the A* algorithm guarantees to return the optimal solution if one exists (Hart et al., 1968).

3.4 Algorithm Details

The A* search algorithm applied in this work is described in Algorithm 2. A *Start* node and the goal condition are first defined to represent the initial state and the goal state of the terrain profile. A set of candidate dragline positions $\mathcal{P}$ is also provided. For each node, the *g_val* represents the cost from the *Start* to this node and the *f_val* is the sum of *g_val* and the heuristic cost from this node to the goal. At each iteration, the algorithm explores all the children of the node that has the minimum *f_val* in *OpenSet*. All its children, which are obtained by applying the digging and dumping operations at the new dragline positions in $\mathcal{P}$ to the current state, are added to *OpenSet*. The *g_val*, *f_val* and *parent* of each child are then updated. The *parent* and the *pos* of a node are used to reconstruct the path from the *Start* to the current node after the search terminates. The algorithm returns failure if no feasible path is found when *OpenSet* is empty.

Several improvements are made to accelerate the searching process. The first improvement results from specifying the maximum searching depth in the search tree. If the node chosen from *OpenSet* reaches the maximum depth, its children will not be explored. This process can be implemented between Line 11 and 12 in Algorithm 2. The second improvement is to forbid duplicate dragline positions in the solution path. That means with increasing searching depth, the number of the candidate positions decreases. The number of iterations required in the loop from Line 12 to 20 in Algorithm 2 is therefore reduced. The final improvement is to prune the children when the sum of the moving and preparation cost is higher than the swing cost in the transition from their parent. This improvement is made by removing these children from *OpenSet* after Line 16. The algorithm was written in C++ and tested on a desktop with a 3.4GHz CPU and 16.0GB RAM. Multiple threads are used when exploring the children of the current node as these process are independent of each other.

Algorithm 2 A* search in dragline positioning planning

Input $Start, Goal, \mathcal{P}$
Output A dragline positioning sequence
1: $Start.pos = null$;
2: $Start.parent = null$;
3: $Start.g\_val = 0$;
4: $Start.f\_val =$ **Heuristic**($Start$, $Goal$);
// candidate nodes for exploration
5: $OpenSet = \{Start\}$;
6: **while** $OpenSet$ is not empty **do**
7: $Current =$ **NodeMinF**($OpenSet$);
8: **if** **Terminate**($Current$, $Goal$) **then**
9: **return** **ReconstructPath**($Current$);
10: **end if**
11: $OpenSet.Remove(Current)$;
12: **for** each candidate position $Pos \in \mathcal{P}$ **do**
13: $[a_g, a_p] =$ **DigDumpStrategy**($Current$,Pos);
14: $Child =$ **GetChild**($Current$,a_g,a_p);
15: $OpenSet.Add(Child)$;
16: $tmp\_g\_val = Current.g\_val +$ **Cost**($Current$, $Child$);
17: $Child.pos = Pos$;
18: $Child.parent = Current$;
19: $Child.g\_val = tmp\_g\_val$;
20: $Child.f\_val = tmp\_g\_val +$ **Heuristic**($Child$, $Goal$);
21: **end for**
22: **end while**
23: **return** failure.

4. SIMULATION RESULTS

The A* algorithm was applied to plan the dragline positioning sequences in two test scenarios that are created based on the actual mining strip shown in Figure 2, which we refer to as Scenarios 1 and 2. The terrain is decomposed and represented in height grids of $2m \times 2m$ cells. Two sets of candidate dragline positions are defined for these two scenarios. The distance between two candidate positions in each set is at least $5m$. Detailed configurations of these two scenarios can be seen in Table 1. We artificially increase the depth of the overburden in Scenario 2 to evaluate the performance of the A* algorithm in a tougher situation. Figures 7 and 8 show the resulting positioning sequences along with the corresponding material movement in both scenarios. For illustration, only a portion of the terrain in Figure 2 is shown in these figures. The digging areas are indicated using a magenta polygon and the overburden is allowed to be dumped in the area to the left of the yellow line. We terminate the search if more than 70% of the coal seam has been exposed in the digging area given its current terrain and the target terrain of the strip. More implementation details of the A* algorithm can be seen in Table 2.

Table 1. Configuration of Scenario 1 and 2. The second column to the last column are: the dragline moving speed, the dragline operating radius, the size of digging area, the number of candidate positions and the overburden depth.

Scenario	v	r	A	N	d_{ob}
1	$4m/min$	$88.4m$	$3480m^2$	23	$16m$
2	$4m/min$	$88.4m$	$4382m^2$	39	$25m$

The operation time of the resulting positioning sequences from the A* algorithm is compared to those obtained by the greedy and the exhaustive enumeration method in Table 3. The greedy method simply chooses the next dragline position with the most removed material given the current terrain at each step. The exhaustive enumeration generates all possible positioning sequences given the maximum searching depth and selects the one that leads to the goal state with the minimum operation time.

Table 2. Implementation details of A*. The last column is the maximum number of positions allowed in the solutions.

Scenario	t_{prep}	$t_{h,sw}$	N_{max}
1	600s	42s	3
2	600s	42s	3

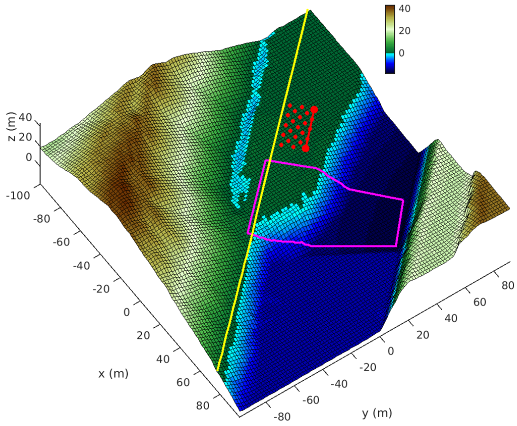
Table 3. Performance of the A* algorithm. The first and the last two columns show overall operation time of the solutions from the A*, the greedy, and the exhaustive search algorithms. The second to the fourth columns are the computation time, the total number of nodes, and the number of explored nodes of the A* algorithm.

Scenario	t_{op,a^*}	t_{cp}	M_{tot}	M_{exp}	$t_{op,g}$	$t_{op,exh}$
1	10268.5s	13.3s	11156	2692	14269.0s	10268.5s
2	39618.1s	532.9s	56356	34933	43985.9s	39618.1s

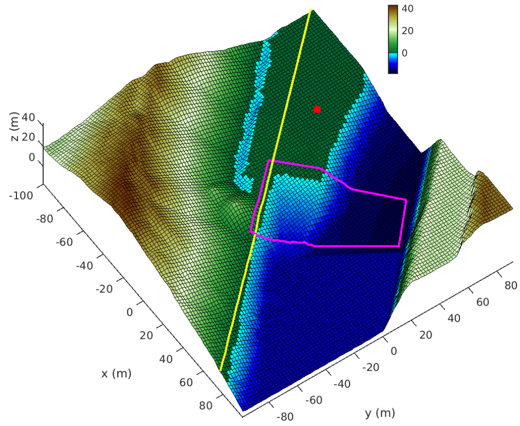
5. CONCLUSION AND FUTURE WORK

This paper is concerned with planning dragline operations at the strategic level. We address the problem of how to position the dragline in order to finish the excavation task of a designated area in a minimum amount of time, when a specified digging and dumping strategy is applied at each position. We use a digging and dumping strategy that indicates the dragline to remove all the material it can reach at a given position and dump the material successively to the nearest available dumping location. In the simulation study, we show that the proposed A* algorithm is capable of generating minimum-cost feasible positioning sequences for two test scenarios. The computation time of these solutions is on the order of minutes, making the algorithm suitable for real-time decision support when incorporated within a receding horizon framework.

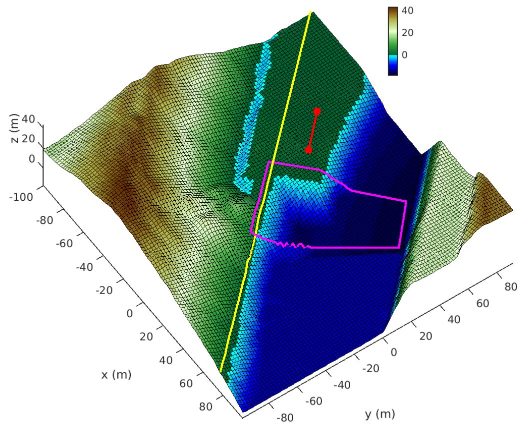
In the future, we look to optimize the entire top-level excavation strategy by including the digging and dumping decisions into the search tree. This tree would likely to be intractable for A*, and we are investigating the potential of using some sampling-based algorithms, such as Monte-Carlo Tree Search (Chaslot et al., 2008). The major issues



(a) Initial terrain.



(b) Terrain after dragline operations at the first position.



(c) Terrain after dragline operations at the first and the second position.

Fig. 7. Test scenario 1 for the A* algorithm. The first figure shows the initial state of the terrain, the candidate positions (small red dots) and the resulting positioning sequence (large red dots with line connection). The rest of the figures show the material movement from the digging area to the dumping area after the dragline operations at each position in the solution sequence.

in this future work are envisaged to be how best to parameterize the digging and dumping decisions, and how to balance between exploration and exploitation in searching the tree. During the research, we have observed that the overall completion time of the excavation task seems to be more sensitive to the dragline positioning sequence than to the digging/dumping decisions. Also, the digging and dumping strategy applied in this work attempts to reduce the overall swing time, which is a dominant component in dragline dig-dump cycles (Humphrey and Wagner, 2011). Therefore, the resulting positioning pattern in this paper could be helpful in guiding the search towards the near-optimal part of the tree in the future algorithm.

REFERENCES

Chaslot, G., Bakkes, S., Szita, I., and Spronck, P. (2008). Monte-carlo tree search: A new framework for game ai. In *AIIDE*.

Corke, P., Winstanley, G., Dunbabin, M., and Roberts, J. (2006). Dragline automation: Experimental evaluation through productivity trial. In *Field and Service Robotics*, 459–468. Springer.

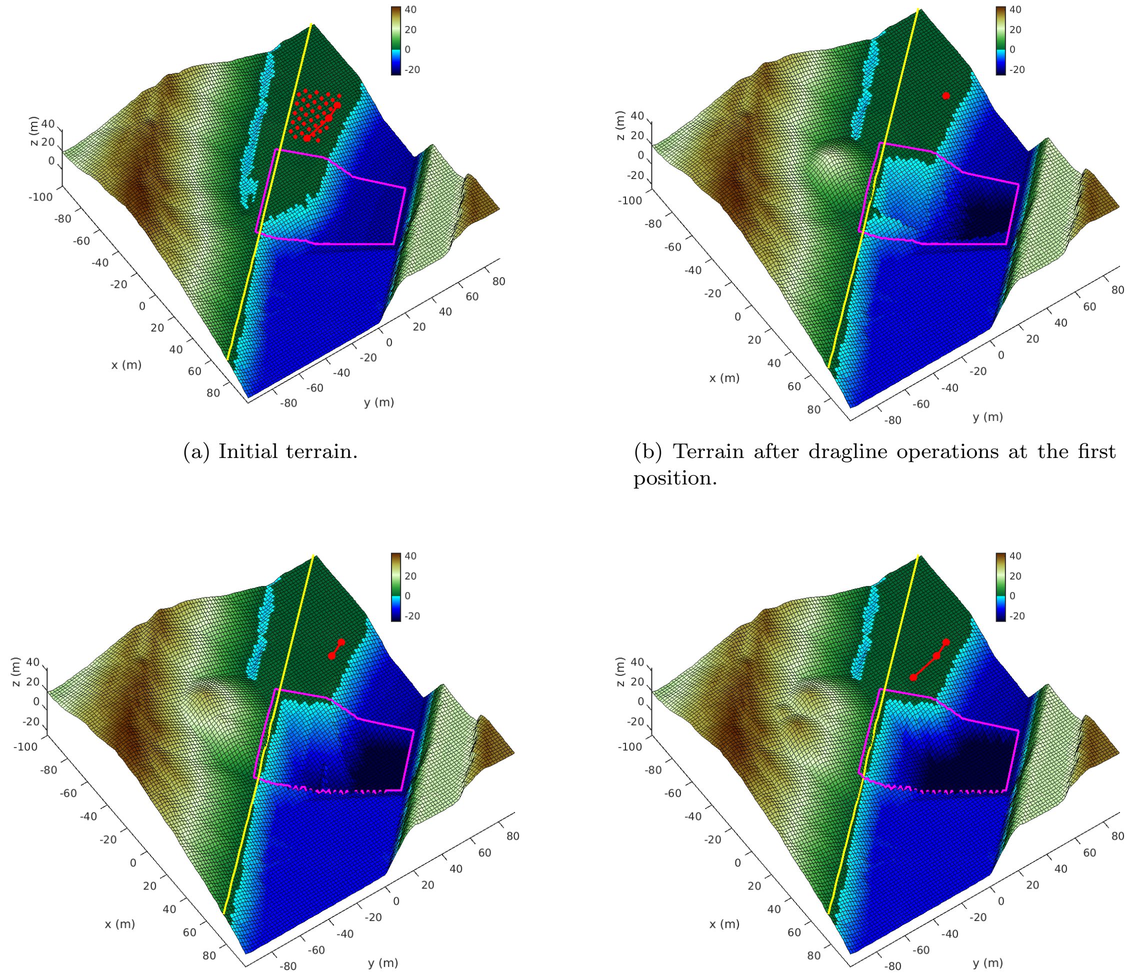
Corke, P.I., Winstanley, G.J., and Roberts, J.M. (1997). Dragline modelling and control. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on Robotics and Automation*, volume 2, 1657–1662. IEEE.

Curragh Queensland Mining Limited (2012). Curragh dragline. `https://en.wikipedia.org/wiki/File:Curragh_Dragline_3.JPG`. Accessed: 2015-04-13.

Hansen, E.A. and Zilberstein, S. (2001). Lao: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1), 35–62.

Hart, P.E., Nilsson, N.J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2), 100–107.

Humphrey, J. and Wagner, J. (2011). Mechanical excavation, loading, and hauling. In P. Darling (ed.), *SME mining engineering handbook*, volume 1, chapter 10.3,



(a) Initial terrain.

(b) Terrain after dragline operations at the first position.

(c) Terrain after dragline operations at the first and the second position.

(d) Final terrain.

Fig. 8. Test scenario 2 for the A* algorithm. The first figure shows the initial state of the terrain, the candidate positions (small red dots) and the resulting positioning sequence (large red dots with line connection). The rest of the figures show the material movement from the digging area to the dumping area after the dragline operations at each position in the solution sequence.

903–929. SME.

Koenig, S., Likhachev, M., and Furcy, D. (2004). Lifelong planning a. *Artificial Intelligence*, 155(1), 93–146.

Li, Z., Zhu, Q., and Gold, C. (2005). *Digital terrain modeling: principles and methodology*, chapter 4, 65–83. CRC Press Inc, New York.

Liu, H., Kearney, M., and Austin, K. (2016). Development of dragline excavation model for operation planning. In *Australasian Conference on Robotics and Automation*.

Mirabediny, H. (1998). *A dragline simulation model for strip mine design and development*. Ph.D. thesis, Department of Civil and Mining Engineering-Faculty of Engineering, University of Wollongong.

Sier, D. (1993). Optimised dragline planning model. Technical report.

Szer, D., Charpillet, F., and Zilberstein, S. (2005). Maa*: A heuristic search algorithm for solving decentralized pomdps. In *21st Conference on Uncertainty in Artificial Intelligence-UAI'2005*.

Thornton, D. and Whiten, B. (2003). Optimization of dragline excavation. Technical report, Julius Kruttschnitt Mineral Research Centre.

Winstanley, G., Usher, K., Corke, P., Dunbabin, M., and Roberts, J. (2007). Field and service applications-dragline automation-a dedade of development-shared autonomy for improving mining equipment productivity. *Robotics & Automation Magazine, IEEE*, 14(3), 52–64.

Winstanley, G.J., Corke, P.I., and Roberts, J.M. (1997). Dragline swing automation. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on Robotics and Automation*, volume 3, 1827–1832. IEEE.