Analysis of Cluster Formation Techniques for Multi-Robot Task Allocation using Sequential Single-Cluster Auctions

Bradford Heap and Maurice Pagnucco

ARC Centre of Excellence in Autonomous Systems
School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW, 2052, Australia.
{bradfordh,morri}@cse.unsw.edu.au

Abstract. Recent research has shown the benefits of using K-means clustering in task allocation to robots. However, there is little evaluation of other clustering techniques. In this paper we compare K-means clustering to single-linkage clustering and consider the effects of straight line and true path distance metrics in cluster formation. Our empirical results show single-linkage clustering with a true path distance metric provides the best solutions to the multi-robot task allocation problem when used in sequential single-cluster auctions.

1 Introduction

We consider a team of autonomous mobile service robots operating in an office-like environment. These robots may be required to deliver mail between rooms, provide an escort to visitors, or complete cleaning tasks. In all of these situations a set of tasks is to be completed and it is our desire that the tasks are distributed amongst all available robots in a manner that seeks to optimise a global team objective.

Recent research has shown market-based sequential auctions can quickly generate good solutions to this class of problem. In particular, *Sequential single-item auctions* (SSI auctions) which allocate tasks to robots one task per auction round at a time provide solutions within guaranteed bounds [11, 8]. Further refinements of this approach have considered various components such as considering complete task allocations with *rollouts* [20], changing the winner determination rules [10], and exchanging tasks post-initial allocation [19, 14].

However, a drawback of allocating tasks one at a time in a market-based environment is that robots are generally greedy in their bidding strategies and will often only consider tasks that seek to minimise their overall cost, rather than the global team cost. For instance, it is common for two robots to be allocated one task each when the overall team cost would be lower if one robot completed these two tasks and the other robot was allocated and completed other tasks.

To overcome this problem, Koenig *et al.* [9] consider an extension of SSI auctions where robots form and bid for combinations of multiple tasks during each auction round. However, despite this approach improving the final task allocation, the calculations required to form the task bundles are very computationally expensive. Heap and Pagnucco considered the merits of this approach in their work on *Sequential single-cluster auctions* (SSC auctions) [5, 6] which uses K-means clustering to form clusters of tasks that are allocated to robots as fixed bundles.

Furthermore, K-means clustering has also been used in a variety of other multi-robot task allocation problems. These include evenly balancing task allocation between robots [4], ensuring robots are spread out in the exploration of unknown space [17, 15], and for map segmentation in RoboCup Rescue Agent Simulation [13]. However, few papers have considered alternative algorithms for the formation of task clusters. In this paper we use SSC auctions to compare K-means clustering to *single-linkage clustering* and consider both straight line distance and true path distance (which takes into consideration obstacles between tasks) as metrics in cluster formation.

In the remainder of this paper we define the multi-robot task allocation problem in the domain of auction-like algorithms, we define SSC auctions, outline each clustering technique, consider the time required for each clustering algorithm to complete, and report our empirical experimental results for each clustering techinque when used in SSC auctions for task allocation. Our key results show single-linkage clustering with a true path distance metric is the best performing clustering technique when used in SSC auctions to solve the multi-robot task allocation problem. However, we also show that the time required to form clusters using a true path distance metric is around 100 times slower than using straight line distances.

2 Multi-Robot Routing

Multi-robot routing (Fig. 1) is considered the standard testbed for the multi-robot task allocation problem in which each task is represented as a location to visit [3]. We follow Koenig *et al.* [9] in their formalisation of the problem. Given a set of robots $R = \{r_1, \ldots, r_m\}$ and a set of tasks $T = \{t_1, \ldots, t_n\}$, any tuple $\langle T_{r_1}, \ldots, T_{r_m} \rangle$ of pairwise disjoint bundles $T_{r_i} \subseteq T$ and $T_{r_i} \cap T_{r_j} = \emptyset$ for $i \neq j$, for all $i = 1, \ldots, m$, is a partial solution of the multi-robot task allocation problem. This means that robot r_i performs the tasks T_{r_i}, and no task is assigned to more than one robot. To determine a complete solution we need to find a partial solution $\langle T_{r_1} \ldots T_{r_m} \rangle$ with $\cup_{r_i \in R} T_{r_i} = T$, that is, where every task is assigned to exactly one robot.

Robots have perfect localisation and can calculate the costs to travel between locations. We assume costs are symmetric, $\lambda(i, j) = \lambda(j, i)$ and are equal across all robots. The robot cost $\lambda_r(T_r)$ is the minimum cost for an individual robot r to visit all locations T_r assigned to it. There can be positive synergies between

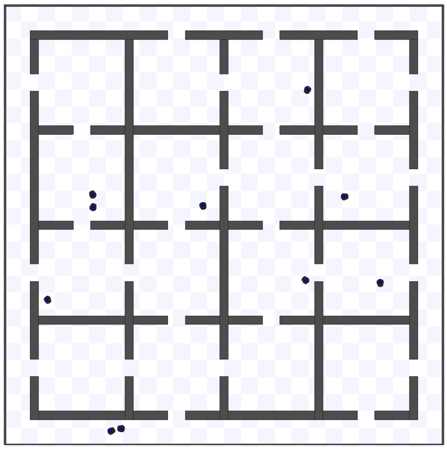


Fig. 1: Multi-Robot Routing

tasks where $\lambda_{r_i}(T_{r_i} \cup T_{r'_i}) < \lambda_{r_i}(T_{r_i}) + \lambda_{r_i}(T_{r'_i})$. Furthermore, we desire to find a complete solution that seeks to minimise a global team objective. In this paper we test with two common team objectives first introduced in [18]:

MiniMax $\max_{r \in R} \lambda_r(T_r)$, that is to minimise the maximum distance each individual robot travels.

MiniSum $\sum_{r \in R} \lambda_r(T_r)$, that is to minimise the sum of the paths of all robots in visiting all their assigned locations.

3 Sequential Single-Cluster (SSC) Auctions

SSC auctions [5] are an extension of SSI auctions and assign fixed clusters of tasks to robots over multiple bidding rounds. At the conclusion of each bidding round one previously unassigned task cluster $c = \{t_1, \ldots, t_o\}$ is assigned to the robot that bids the least for it so that the overall team cost increases the least. After all task clusters are allocated, each robot seeks to complete all its allocated tasks in as short a distance possible. Robots do not have to do all tasks in a cluster sequentially. When a robot is awarded a new cluster, the robot adds the tasks in this new cluster to its existing task assignment and replans its path to travel.

We formulate the algorithm for SSC auctions in Fig. 2. Each robot runs the algorithm independently of other robots and, with the exception of supplying the initial list of tasks and clusters to each robot, there is no centralised controller. Before the SSC auction algorithm begins, a clustering algorithm is used to allocate all tasks into task clusters and $C = \{c_1, \ldots, c_k\}$ is the set of all clusters. Each task is assigned to one, and only one cluster, and clusters can be of varying sizes. All robots are informed of all tasks and all clusters.

function SSC-Auction ($\bar{C}$,C_r, r, R)
Input: $\bar{C}$: the set of clusters to be assigned
C_r: the set of clusters presently assigned
r: the robot r
R: the set of robots R
Output: C_r: the set of clusters assigned to the robot

1: **while** ($\bar{C} \neq \emptyset$)
2: /* *Bidding Stage* */
3: **for each** cluster $c \in \bar{C}$
4: $\beta_r^c \leftarrow$CalcBid(C_r,c);
5: Send(β_r^c, R) | $B \leftarrow \bigcup_i$Receive($\beta_{r_i}^c$, R);
6: /* *Winner-Determination Stage* */
7: $(r', c) \leftarrow \arg\min_{(r' \in R, c \in \bar{C})} B$;
8: **if** $r = r'$ **then**
9: $C_r \leftarrow C_r \cup \{c\}$;
10: $\bar{C} \leftarrow \bar{C} \backslash \{c\}$;

Fig. 2: Sequential Single-Cluster Auctions

The SSC auction begins and continues while there are unassigned task clusters (Line 1). During the bidding stage (Lines 2-5) the robot calculates bids for every unassigned task cluster and submits its lowest bid to all other robots. Each bid calculation requires robots to provide a solution to the travelling repairman problem [1]. Because this problem is NP-hard, robots often use the cheapest-insertion and two-opt heuristics [2] to provide a close approximation to the optimal solution. Each bid is a triple $\beta = \langle b_r, b_c, b_\lambda \rangle$ of a robot b_r, a task cluster b_c and a bid cost b_λ. The function CalcBid takes the set of previously assigned clusters C_r to robot r and the cluster c being bid on and uses a bidding rule to calculate a bid cost (Line 4). The robots send their bids and receive all bids from other robots in parallel (Line 5). The winner-determination stage (Lines 6-10) consists of each robot choosing the task cluster with the lowest bid from the set of submitted bids. Ties are broken in an arbitrary manner. The robot with the winning bid has the winning task cluster assigned to it. All robots then remove the winning task cluster from the set of unassigned clusters and the next bidding round begins.

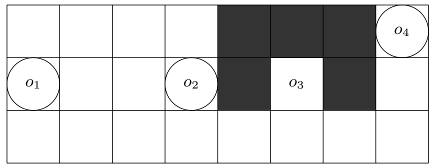
4 Clustering Techniques

We now consider two different models of cluster formation. K-means clustering is a form of centroid based clustering where each object is assigned to a single cluster based on the object's proximity to the centre of the cluster. Single-linkage clustering is a form of connectivity based clustering which recursively merges clusters by the minimum distance between two objects in each cluster. In this

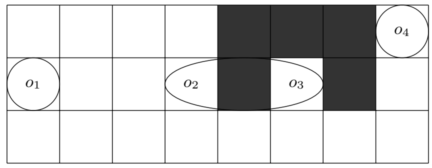
function K-means (T,k)
Input: T: the set of tasks to be clustered
k: the number of clusters to form
Output: C: the set of clusters

1: $M \leftarrow$ InitialiseClusterCentres(T,k);
2: **while** *cluster centres have changed*
3: /* *Task Cluster Assignment Stage* */
4: **for each** task $t \in T$
5: **for each** cluster centre $m_i \in M$
6: $\lambda^t_{m_i} \leftarrow$CalcDistance($t$,$m_i$);
7: $C_{min_{(\lambda^t_{m_i})}} \leftarrow t$;
8: /* *Update Cluster Centres Stage* */
9: **for each** cluster $c \in C$
10: $M_c \leftarrow$CalcCentre(c);

Fig. 3: K-means Clustering



a: Initial Cluster Centres



b: Stable Cluster Centres

Fig. 4: Formation of three clusters of four objects using K-means clustering with a straight line distance metric.

section we also compare the effects of using straight line distance and true path distance as metrics for both clustering models.

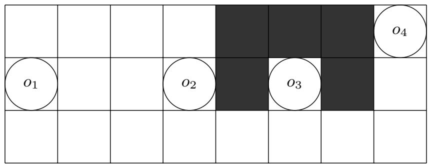
The standard K-means clustering algorithm [12] is given in Fig. 3 and an example cluster formation with a straight line distance metric is presented in Fig. 4. Before the algorithm begins the initial centres of all clusters must be selected (Line 1). A common approach for this is to randomly select k objects from the set of data to be clustered and use each of these objects as an initial cluster centre (Fig. 4a). The algorithm then alternates between two stages until the membership of all clusters is stable (Fig. 4b). During the task cluster assignment stage (Lines 3 - 7) every task is considered independently. The distance between the task and every cluster centre is calculated and the task is reassigned from its current cluster to the cluster with the minimum distance to itself. During the update cluster centres stage (Lines 8 - 10) the centre of each cluster is recalculated to reflect the changes in the membership of each cluster. The algorithm then repeats until no object moves between clusters.

We present an algorithm to perform single-linkage clustering [16] in Fig. 5 and give an example cluster formation with a true path distance metric in Fig. 6. The algorithm begins with each object being assigned to a cluster containing only itself (Lines 1 - 2) (Fig. 6a). Clusters are then repeatedly merged until the number of clusters is equal to k (Lines 3 - 10) (Fig. 6b). To merge clusters we calculate the distance between every individual object in each cluster and every object in every other cluster (Lines 4 - 6). The two clusters containing the two objects with the minimum distance between them are then merged (Lines 7 -

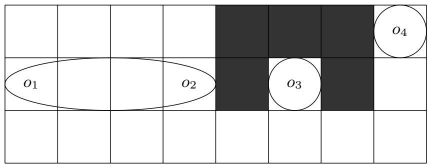
function Single-linkage (T,k)
Input: T: the set of tasks to be clustered
k: the number of clusters to form
Output: C: the set of clusters

1: **for each** task $t_i \in T$
2: $C_i \leftarrow t_i$;
3: **while** $|C| > k$
4: **for each** cluster $C \supset C_i$
5: **for each** cluster $C \backslash C_i \supset C_j$
6: $\lambda_{C_j}^{C_i} \leftarrow$CalcDistance$(C_i, C_j)$;
7: **if** $min(\lambda_{C_j}^{C_i})$ **then**
8: $C_{merged} \leftarrow \{C_i, C_j\}$;
9: $C \leftarrow C \cup \{C_{merged}\}$;
10: $C \leftarrow C \backslash \{C_i, C_j\}$;

Fig. 5: Single-linkage Clustering



a: Initial Clusters



b: Final Merged Clusters

Fig. 6: Formation of three clusters of four objects using single-linkage clustering with a true path distance metric.

10). The algorithm then repeats until there are k clusters.

In our experiments we expect that the differences in the design of these two algorithms will cause vastly different task allocations to robots. Due to K-means clustering focussing on 2-dimensional areas of tasks, we expect that each robot will be generally constrained to completing tasks within an isolated area. In comparison, task clusters formed using single-linkage clustering are more likely to see robot paths crossing over each other as the task formation is focussed on the 1-dimensional connection between any two tasks.

5 Cluster Formation Time Analysis

The length of time required to formulate clusters is crucial in finding a good solution to the multi-robot task allocation problem. The time complexity for K-means clustering is $O(|T|^{dk+1} \log |T|)$ (where d is the number of dimensions) [7] and for single-linkage clustering is $O(|T|^2)$. Generally speaking single-linkage clustering is much quicker than K-means clustering.

However, it is also important to take into account the time involved in the calculation of the distance metric. When we consider the calculation of a straight line Euclidean distance between objects the time required is minimal. Contrary to this is the time required to calculate the true path distance between two objects taking into account obstacles. Even in a two dimensional environment, such as the map presented in Fig. 4 and Fig. 6, two geographically close objects

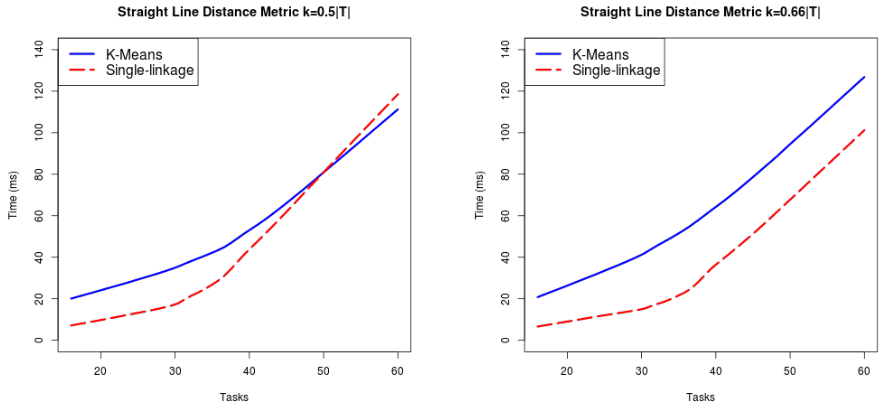


Fig. 7: Cluster Formation Time using a Straight Line Distance Metric

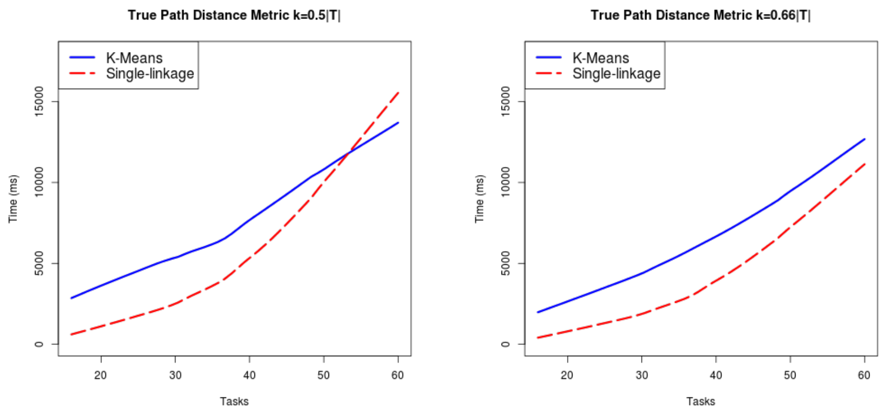


Fig. 8: Cluster Formation Times using a True Path Distance Metric

o_2 and o_3 have a true path distance that is greater than the distance between o_1 and o_2. To calculate the true path distance we are required to perform a search for the shortest distance between every object and every other object using an occupancy grid map. The number of true path calculations required for single-linkage clustering is constrained by the number of tasks. However, in K-means clustering every time a cluster centre is changed we are required to recalculate the distance from the centre to all objects.

To examine the real time requirements of each clustering algorithm we simulate an office-like environment with 16 rooms each containing four interconnecting doors that can be independently opened or closed to allow or restrict travel between rooms (Fig. 1). We test on 25 different fixed configurations of combinations of opened and closed doors. In each configuration we guarantee there is an open path between each room and every other room. For each configuration we test a wide range of total tasks to be clustered $|T| \in \{16, \ldots, 60\}$. For each task set we repeat the clustering process for two different values of k, $k = \frac{1}{2}|T|$ and $k = \frac{2}{3}|T|$. To calculate the true path cost between tasks we use an A* search on an occupancy grid map of each office environment with our heuristic being the straight line distance between the two tasks.

The results of clustering using a straight line distance metric are plotted in Fig. 7 and the results of clustering using a true path distance metric are plotted in Fig. 8. These plots show that generally, as expected, single-linkage clustering completes quicker than K-means clustering. However, when there are a large number of tasks and $k = \frac{1}{2}|T|$ single-linkage clustering takes a long period of time to complete. This is due to the large number of cluster merges required when k is small. We note that K-means clustering does not suffer this problem as the stablisation of clusters is independent of the value of k. Finally, we also observe that the use of a true path distance metric causes both clustering algorithms to perform about 100 times slower compared to the straight line distance metric.

6 Empirical Analysis using SSC Auctions

We now test each of these clustering techniques with SSC auctions to solve the multi-robot task allocation problem for both the MiniMax and the MiniSum team objectives. Using the clusters formed in Sect. 5 we test homogeneous mobile robots in teams of varying sizes $|R| \in \{4, 6, 8, 10\}$. In each of the 25 office configurations robots are initially positioned in different random locations. Robots can only travel between rooms through open doors and they cannot open or close doors. We present the mean results of the maximum distance and the summation of all distances travelled for the two team objectives in Tables 1, 2, 3 and 4.

The results for SSC auctions with robots bidding according to the MiniMax team objective are shown for clusters formed with $k = \frac{1}{2}|T|$ in Table 1 and for clusters formed with $k = \frac{2}{3}|T|$ in Table 2. Both of these result tables show that generally the use of a true path distance metric results in task allocations that have lower mean maximum robot travel distances. Overall, single-linkage clustering with a true path distance metric produces the best task allocations for this team objective and K-means clustering with a straight line metric performs the worst.

To confirm the statistical validity of these results we perform *non-parametric one-sided Wilcoxon signed-rank tests* for each robot/task/cluster combination. We choose this statistical test as we cannot make distribution assumptions due to the differences in robot initial locations and the map configurations of opened and closed doors for each experiment. We seek to confirm that the use of a true path distance metric in cluster formation results in lower final travel distances than clusters formed using a straight line distance metric. Our null hypothesis is defined as $H_0 : \mu\lambda_{true\text{-}path} \geq \mu\lambda_{straight\text{-}line}$ and our alternative hypothesis as $H_0 : \mu\lambda_{true\text{-}path} < \mu\lambda_{straight\text{-}line}$.

Table 1: Mean Maximum Distance for MiniMax Team Objective with $k = \frac{1}{2}|T|$

			Straight Line Metric		True Path Metric	
Robots	Tasks	Clusters	K-means	Single-link	K-means	Single-link
4	16	8	995	954	906	874
6	24	12	983	963	816	777
8	32	16	895	921	773	690
10	40	20	823	870	677	636
4	20	10	1090	1084	989	932
6	30	15	1023	1032	888	833
8	40	20	898	918	802	776
10	50	25	814	883	745	679
4	24	12	1203	1174	1053	1027
6	36	18	1117	1094	965	913
8	48	24	988	983	863	810
10	60	30	887	886	805	690

Table 2: Mean Maximum Distance for MiniMax Team Objective with $k = \frac{2}{3}|T|$

			Straight Line Metric		True Path Metric	
Robots	Tasks	Clusters	K-means	Single-link	K-means	Single-link
4	16	11	910	873	864	874
6	24	16	810	874	750	713
8	32	21	830	840	727	671
10	40	27	718	746	669	617
4	20	13	1004	1043	991	917
6	30	20	932	941	841	826
8	40	27	818	831	762	731
10	50	33	752	753	698	652
4	24	16	1015	1154	1029	967
6	36	24	955	969	936	886
8	48	32	884	863	840	785
10	60	40	860	742	724	698

For K-means clustering with $k = \frac{1}{2}|T|$ we get a significant difference with confidence greater than 0.90 for all robot/task/cluster combinations tested. However, for K-means clustering with $k = \frac{2}{3}|T|$ our results are not significant for all robot/task/cluster combinations. In particular, our mean results for the configuration of 4 robots, 24 tasks, and 16 clusters has true path distances resulting in a worst maximum distance travelled than the use of a straight line metric. We speculate that the cause of these non-significant results is due to K-means clustering seeking to form non-overlapping clusters of geographically close tasks, whereas robots, in seeking to minimise their path travelled, may not confine themselves to particular local geographic areas.

Our results for single-linkage clustering have much stronger statistical significance. For all robot/task/cluster combinations we obtain confidence 0.97 and greater for both $k = \frac{1}{2}|T|$ and $k = \frac{2}{3}|T|$. Finally, we compare our best and worst results of single-linkage clustering with true path distances to K-means clustering with straight line distances. Again for $k = \frac{1}{2}|T|$ we get a very significant

Table 3: Mean Summed Distance for MiniSum Team Objective with $k = \frac{1}{2}|T|$

			Straight Line Metric		True Path Metric	
Robots	Tasks	Clusters	K-means	Single-link	K-means	Single-link
4	16	8	2298	2231	2197	2140
6	24	12	2801	2701	2565	2489
8	32	16	3123	3061	2893	2806
10	40	20	3444	3301	3087	3000
4	20	10	2701	2621	2483	2464
6	30	15	3239	3083	2943	2850
8	40	20	3627	3504	3300	3211
10	50	25	3775	3741	3459	3409
4	24	12	3011	2864	2767	2715
6	36	18	3511	3464	3272	3258
8	48	24	3917	3821	3605	3541
10	60	30	4208	4059	3878	3786

Table 4: Mean Summed Distance for MiniSum Team Objective with $k = \frac{2}{3}|T|$

			Straight Line Metric		True Path Metric	
Robots	Tasks	Clusters	K-means	Single-link	K-means	Single-link
4	16	11	2232	2178	2142	2117
6	24	16	2642	2640	2519	2501
8	32	21	3090	3009	2896	2809
10	40	27	3105	3185	3045	2990
4	20	13	2636	2569	2495	2460
6	30	20	3101	3058	2897	2878
8	40	27	3329	3419	3241	3229
10	50	33	3626	3500	3446	3375
4	24	16	2839	2850	2783	2748
6	36	24	3339	3401	3268	3230
8	48	32	3813	3643	3551	3350
10	60	40	4160	3849	3850	3814

result with confidence 0.99 and with $k = \frac{1}{2}|T|$ we get confidence 0.97. Overall, we can conclude that using true path distance metrics produces much better solutions to the multi-robot task allocation problem than straight line distances for the MiniMax team objective.

Our results for robots bidding according to the MiniSum team objective are likewise shown for $k = \frac{1}{2}|T|$ in Table 3 and for $k = \frac{2}{3}|T|$ in Table 4. This data mirrors our results for the MiniMax team objective with single-linkage clustering using a true path metric producing the best results and K-means clustering with a straight line metric the worst.

Again we perform *one-sided Wilcoxon signed-rank tests* to confirm the statistical validity of our data. For K-means clustering with $k = \frac{1}{2}|T|$ we confirm a very significant result with confidence 0.99. However, when tested with clusters of $k = \frac{2}{3}|T|$ K-means clustering again fails to deliver a strong statistical confidence. This is despite our overall means showing the use of a true path distance metric outperforming a straight line distance metric in all robot/task/cluster combinations.

For single-linkage clustering we again get strong statistically significant results. For clusters formed with $k = \frac{1}{2}|T|$ we measure a confidence result of 0.995 and for $k = \frac{2}{3}|T|$ a confidence of 0.95. Finally, we conclude by testing the significance of the difference between our best performing single-linkage clustering using true path distances and worst performing K-means clustering using straight line costs. Comparing $k = \frac{1}{2}|T|$ we get a confidence of 0.998 and for $k = \frac{2}{3}|T|$ a confidence of 0.95.

In summary, for both MiniMax and MiniSum team objectives tested, we have shown the power of using a true path distance metric in the formation of clusters to solve the multi-robot task allocation problem. Despite our cluster formation time measurements showing that the use of true path distance metrics is around 100 times slower than straight line calculations, we believe that the overall travel distance saved as a result of better clusters far outweighs the extra initial time spent on cluster formation. Futhermore, it can be argued that the cluster formation time with a true path metric will be much smaller than the expected execution time of the robots completing all allocated tasks.

7 Conclusions and Further Work

We have presented an analysis of clustering techniques to solve the multi-robot task allocation problem using SSC auctions. We have considered the time required using two different clustering models and the effect of calculating true path distances instead of straight-line distances in the formation of clusters. Our empirical results show the benefit of using single-linkage clustering with a true-path distance metric over other cluster formation techniques.

There remains much scope for future work. Finding a good value for k remains a challenge. A large k value results in clusters that contain few tasks and, as such, little inter-task synergy is considered. On the other hand, small k value results in clusters containing many tasks can lead to robots being allocated tasks and resultant paths that would be better suited to other robots. A clustering approach that sought a balance between these two challenges would be ideal, however, the time complexity may be much greater than our existing approaches.

In a related vein, disparity between the number of tasks contained in each cluster can lead to clusters containing few tasks being allocated in earlier auction rounds than larger clusters. Future work could consider the effects when robots take into account the size of clusters during the formation of bids. Allocating large clusters first may lead to better solutions than a series of small clusters being allocated in early auctions rounds.

Finally, we also wish to consider more complicated extensions to the multi-robot task allocation problem. Of particular interest is the *courier delivery problem* which requires robots to pick up and drop off objects. Auctioning clusters of tasks in this problem domain is much more difficult as we are required to consider both the pick up and drop off locations of objects when forming clusters.

References

1. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing. pp. 163–171 (1994)
2. Croes, G.: A method for solving traveling-salesman problems. Operations Research 6, 791–812 (1958)
3. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (2006)
4. Elango, M., Nachiappan, S., Tiwari, M.K.: Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms. Expert Systems with Applications 38(6), 6486 – 6491 (2011)
5. Heap, B., Pagnucco, M.: Sequential single-cluster auctions for robot task allocation. AI 2011 LNAI 7106, 412–421 (2011)
6. Heap, B., Pagnucco, M.: Repeated sequential auctions with dynamic task clusters. Proc. AAAI-12 (2012)
7. Inaba, M., Katoh, N., Imai, H.: Applications of weighted voronoi diagrams and randomization to variance-based K-clustering. In: Proceedings of the tenth annual symposium on Computational geometry. pp. 332–339. ACM (1994)
8. Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., Jain, S.: The power of sequential single-item auctions for agent coordination. Proc. AAAI-06 (2006)
9. Koenig, S., Tovey, C., Zheng, X., Sungur, I.: Sequential bundle-bid single-sale auction algorithms for decentralized control. Proc. IJCAI-07 pp. 1359–1365 (2007)
10. Koenig, S., Zheng, X., Tovey, C., Borie, R., Kilby, P., Markakis, V., Keskinocak, P.: Agent coord. with regret clearing. AAAI-08 (2008)
11. Lagoudakis, M., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., Jain, S.: Auction-based multi-robot routing. Proc. Int. Conf. on Robotics: Science and Systems pp. 343–350 (2005)
12. Lloyd, S.: Least squares quantization in pcm. Information Theory, IEEE Transactions on 28(2), 129–137 (1982)
13. Nanjanath, M., Erlandson, A., Andrist, S., Ragipindi, A., Mohammed, A., Sharma, A., Gini, M.: Decision and coordination strategies for robocup rescue agents. Simulation, Modeling, and Programming for Autonomous Robots pp. 473–484 (2010)
14. Nanjanath, M., Gini, M.: Repeated auctions for robust task execution by a robot team. Robotics and Autonomous Systems 58(7), 900–909 (2010)
15. Puig, D., Garcia, M., Wu, L.: A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation. Robotics and Autonomous Systems (2011)
16. Sokal, R., Sneath, P., et al.: Principles of numerical taxonomy. Principles of numerical taxonomy. (1963)
17. Solanas, A., Garcia, M.: Coordinated multi-robot exploration through unsupervised clustering of unknown space. In: Proc. IROS-04. pp. 717–721 (2004)
18. Tovey, C., Lagoudakis, M., Jain, S., Koenig, S.: The generation of bidding rules for auction-based robot coordination. Multi-Robot Systems. From Swarms to Intelligent Automata Volume III pp. 3–14 (2005)
19. Zheng, X., Koenig, S.: K-swaps: Cooperative negotiation for solving task-allocation problems. In: Proc. IJCAI-09. pp. 373–379 (2009)
20. Zheng, X., Koenig, S., Tovey, C.: Improving sequential single-item auctions. Proc. IROS-06 pp. 2238–2244 (2006)