IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

An Approximation Algorithm for a Task Allocation, Sequencing and Scheduling Problem involving a Human-Robot Team

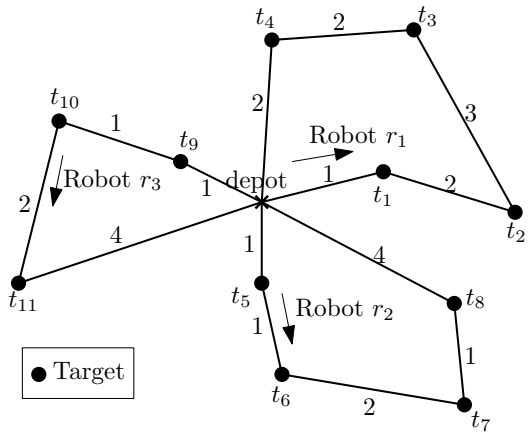
Sai Krishna Kanth Hari[1], Abhishek Nayak[1], and Sivakumar Rathinam[1]

***Abstract*— This article presents an approximation algorithm for a Task Allocation, Sequencing and Scheduling Problem (TASSP) involving a team of human operators and robots. The robots have to travel to a given set of targets and collaboratively work on the tasks at the targets with the human operators. The problem aims to find a sequence of targets for each robot to visit and schedule the tasks at the targets with the human operators such that each target is visited exactly once by some robot, the scheduling constraints are satisfied and the maximum mission time of any robot is minimum. This problem is a generalization of the single Traveling Salesman Problem and is NP-Hard. Given k robots and m human operators, an algorithm is developed for solving the TASSP with an approximation ratio equal to $\frac{5}{2} - \frac{1}{k}$ when $m \geq k$ and equal to $\frac{7}{2} - \frac{1}{k}$ otherwise. Computational results are also presented to corroborate the performance of the proposed algorithm.**

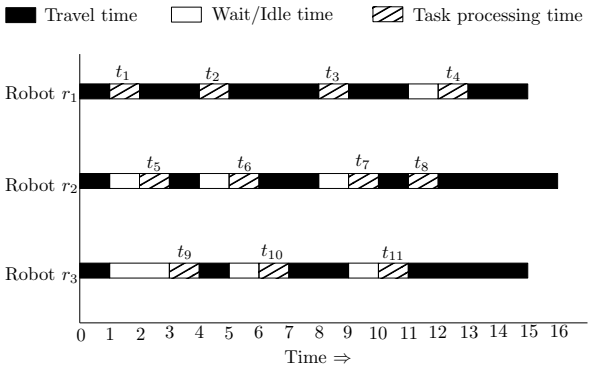
I. Introduction

This article considers a mission planning problem which arises when human operators located at a base station have to collaboratively work with a team of mobile robots to visit a set of targets and perform inspection, classification or data collection tasks at the targets. Each target is associated with a task which must be collaboratively performed by both a robot and a human operator. Each human operator is only allowed to work on at most one task and each task requires at most one human operator to work on it at any time instant. The robots travel to the targets, collaboratively work with the human operators to complete the tasks at the targets and return to their initial position (depot). The allocation of targets (or tasks) and the sequence in which the targets must be visited for each robot is not known apriori. Also, when the number of human operators is less than the number of robots, it is possible that a robot has to wait at a target for a human operator to be available to collaboratively work on the task. The mission time of each robot will include its travel time, wait time and the processing times of the tasks at the targets assigned to it. The objective of the problem is to find a sequence of targets (tasks) for each robot to visit and schedule its tasks with the human operators such that each task is performed once by some robot, the scheduling constraints are satisfied and the maximum mission time of any of the robots is minimum. This problem is referred to as the Task Allocation, Sequencing and Scheduling Problem (TASSP). Refer to Fig. 1 for an illustration of a feasible solution to the TASSP where a team of 3 robots and 1 human operator are required to complete the tasks at 11 targets. Fig.

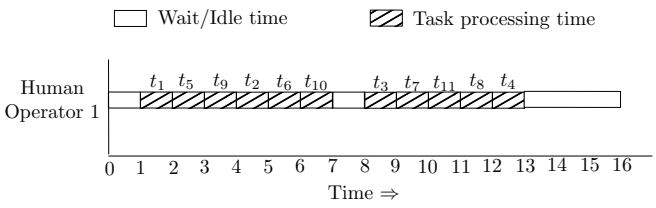
[1] All the authors are part of the Department of Mechanical Engineering, Texas A&M University, College Station, TX, 77843 (e-mail: srathinam@tamu.edu)



(a) The routes followed by the 3 robots (denoted by r_1, r_2, r_3). The depot denotes the initial/final location of all the robots. Targets are denoted as $t_1, \cdots, t_{11}$. The travel times are also shown adjacent to their corresponding travel edges.



(b) The Gantt chart for the robots corresponding to the routes shown in Fig. 1(a). The processing time of the task at any target is 1 unit. The mission time of each robot consists of its travel time (black bars), wait/idle time (white bars) and task processing time (green bars). As there is only one human operator, at most one robot can work on a task with the operator at any time instant.



(c) The Gantt chart for the human operator corresponding to the robot schedules shown in Fig. 1(b).

Fig. 1. A feasible solution to the TASSP. There are 11 targets, 3 robots and 1 human operator in this example.

1(a) shows the allocation of targets and the route of each robot. Figs. 1(b), 1(c) show the corresponding Gantt charts for the robots and the human operator.

TASSP naturally arises in data gathering or surveillance applications involving unmanned systems [1] where few human operators are expected to collaboratively work with a large team of robots to perform specific tasks at the targets. Incorporating human factor requirements in unmanned vehicle routing has been of significant interest in military applications over the last decade [2], [3]. As discussed in the survey [4], human operators in some unmanned applications should be allowed sufficient time to examine the real-time information at the target locations visited by the vehicles. In-site inspection and target classification applications where human input is critical require robots to interact with a human operator to classify the targets accurately in real-time [5]. Robots may also have manipulators that need to be tele-operated by a human operator. Specifically, in crop monitoring applications [6], the cameras on a robot are tele-operated by a human operator to get the images with a desired resolution and field of view.

If the processing times of the tasks at the targets are negligible and if there is only one robot, TASSP reduces to a single Traveling Salesman Problem (TSP). Therefore, TASSP is a generalization of the TSP and is NP-Hard. Hence, we are interested in developing approximation algorithms for the TASSP. An α-approximation algorithm for an optimization problem is a polynomial time algorithm that produces a feasible solution whose cost is at most α times the optimal cost for any instance of the problem. The factor α, which indicates the theoretical guarantee on the quality of the solutions obtained, is also referred to as the approximation ratio. Developing approximation algorithms are very useful because they can quickly provide feasible solutions to NP-Hard problems with *a-priori* guarantees, $i.e.$ guarantees that are obtained prior to implementing the algorithm on an specific instance of the problem. The solution provided by an approximation algorithm can also be used as an initial solution for meta-heuristics and further be improved upon [7], or can be used to warm-start the Branch and Cut solvers as done in [8]. Note that approximation ratios are worst-case guarantees true for any instance of the problem; the quality of a solution obtained by implementing the approximation algorithm for a specific instance is generally better than the guarantee indicated by the approximation ratio[1]. Currently, there are no approximation algorithms known for the TASSP. The following are the main contributions of this article:

1) Given k robots and m human operators, an algorithm is developed for solving the TASSP with an approximation ratio equal to $\frac{5}{2} - \frac{1}{k}$ when $m \geq k$ and equal to $\frac{7}{2} - \frac{1}{k}$ otherwise. The approximation algorithm modifies the travel times of the robots to also include the processing times of the tasks at the targets, and combines a well known multiple TSP algorithm [9] with a greedy heuristic. The key contribution here is in the proof of the approximation ratio.
2) Numerical results are presented to corroborate the performance of the approximation algorithm on a large set of test instances. For small instances, optimal costs (obtained by solving an integer program) are used to infer the quality of solutions produced by the approximation algorithm. For large instances, lower bounds are used as proxies for the optimal cost to infer the quality of solutions obtained by the approximation algorithm.

[1]In Section VI, we further provide numerical results to illustrate this issue further.

II. Literature Review

TASSP is closely related to a general class of routing and scheduling problems involving both vehicles and crews which have numerous variants based on the application [10], [11]. Generally in crew scheduling, a human operator is assigned to a vehicle, and is tasked to both drive the vehicle and aid in the delivery or pickup of goods at target locations. On the other hand, in the TASSP, a human operator remotely works with a robot (vehicle) only while collaborating on the task at a target location; after the task is completed, the robot travels to its next destination without the aid of the human operator, and the human operator is available to work on any pending tasks with other robots. TASSP naturally arises in human-robot collaborations as presented in [2], [3]. In this section, we review the relevant literature in the area of unmanned vehicles, TSPs and scheduling to provide a context for our contributions to the TASSP.

A. Related work in unmanned vehicles

In [12], the authors consider a task allocation problem for a heterogeneous team of human operators and robots. They plan schedules based on a distributed approach while taking into account the workload on the human operators, the availability of robots and some coordination constraints. In [13], the authors address a joint task allocation and scheduling problem with coupled temporal and spatial constraints in the context of manufacturing applications. They present fast task sequencing heuristics and mixed integer linear programming based methods to find near-optimal solutions for real-world problems. In [14], a joint robot routing/scheduling problem is considered for reducing the human operator's workload and the loitering times of the robots. The problem is first formulated as a mixed integer nonlinear program and then transformed to a mixed integer linear program, and based on these programs, a dynamic solution strategy is proposed to find sub-optimal schedules and routes.

A very generic version of the TASSP with heterogeneous vehicles, tasks with time window constraints and human operator specific scheduling constraints is presented in [2]. The authors in [2] also present a mathematical programming model and demonstrate the effectiveness of including human operator constraints in planning the routes of the vehicles. In [5], a collaborative human-robot TSP is formulated where the objective is to maximize the target classification probability for a single robot while deciding on the targets that needs to

be visited by the robot, the targets that need human assistance and the targets that can be classified by remote sensing.

B. Related work in Routing/TSPs

When the travel costs are symmetric and satisfy the triangle inequality, there is a $\frac{3}{2}$-approximation algorithm by Christofides [15] for the single TSP. For the multiple robot case, generally there are two objectives considered in the literature; the first objective (also referred to as *min-sum*) aims to minimize the sum of the travel costs of all the robots (or vehicles) and the second objective (also referred to as *min-max*) aims to minimize the maximum travel time of any robot. There are several approximation algorithms known for the *min-sum* objective [16]–[18]. The *min-max* objective, which is harder to deal with, is a special case of the TASSP when the processing times of all the tasks at the targets are equal to 0. Specifically, in the *min-max* multiple TSP, the mission time of each robot is incurred only due to travel; the objective is to find a sequence of targets to visit for each robot such that each target is visited once by some robot, the robots return to their initial location (or the depot) after visiting the targets, and the maximum travel time of any of the robots is minimum. Frederickson et al. [9] developed the first approximation algorithm for the *min-max* multiple TSP. This algorithm first finds a sequence of targets by solving a single TSP using the Christofides algorithm and then splits the sequence into k sub-sequences (one for each robot) of more or less equal travel times. Frederickson et al. [9] proved the approximation ratio of their algorithm is $\frac{5}{2} - \frac{1}{k}$ if the travel times are symmetric and satisfy the triangle inequality. No algorithms with approximation ratios better than $\frac{5}{2} - \frac{1}{k}$ have been known yet for the *min-max* multiple TSP.

C. Related work in Scheduling

If there is no travel involved for the robots (say, all the targets are at the depot itself) and the objective is to schedule the tasks with the human operators such that no operator works on more than one task at any time instant and the maximum completion time of any task must be minimized, the TASSP reduces to a Parallel Machine Scheduling Problem (PMSP) [19] which is also NP-Hard. In TASSP, the sequence of targets visited by each robot influence how the corresponding tasks are scheduled with the human operators.

A greedy heuristic was presented for solving the PMSP by Graham [19] with an approximation ratio of $2 - \frac{1}{m}$ where m denotes the number of human operators (or machines). There are also algorithms with better approximation ratios [19], [20] known for the PMSP, but nevertheless, the greedy heuristic by Graham [19] is simple to implement and works also when the tasks arrive in an online setting.

The TASSP is a generalization of the PMSP and the *min-max* multiple TSP. Currently, there is no approximation algorithm in the literature available for solving the TASSP. In the next section, we formally introduce the TASSP. In section IV, we present the approximation algorithm for the TASSP and prove its theoretical guarantee in section V. Numerical results are then presented in section VI to corroborate the performance of the proposed algorithm.

III. Problem Statement

Let $T := \{t_1, t_2, \cdots, t_n\}$ denote the set of targets and $R := \{r_1, r_2, \cdots, r_k\}$ represent the robots. Assume there are at least two robots ($k \geq 2$), and all of them are initially located at the depot d. Let $V := T \cup \{d\}$. Let $c(i, j)$ denote the time required to travel between any two vertices $i, j, \in V$. The travel times are symmetric and satisfy the triangle inequality. That is, $c(i, j) = c(j, i)$ and $c(i, j) \leq c(i, k) + c(k, j)$ for all $i, j, k \in V$.

Each target has a specific task that needs to be performed collaboratively between a robot and a human operator. We assume each task must be completed without preemption, $i.e.$, once a human operator starts working with a robot on a task, the task cannot be interrupted before it it processed completely. Suppose $p_i > 0$ denotes the processing time needed to collaboratively complete the task at target $i \in T$. There are m human operators available to perform the tasks and it is assumed that each human operator can handle at most one task at any time instant.

A robot on arrival at a target may have to wait at the target if there is no human operator available to work with the robot, $i.e.$, all the human operators are busy working with other robots. In this case, the robot waits for the next available human operator to process the task. Therefore, the total mission time incurred by a robot is the sum of its travel time, wait time and the processing times of all the tasks at the targets visited by it. A schedule for a robot is a partitioning of its mission time into time intervals where each time interval corresponds to its travel time or wait time or task processing time.

The objective of the problem is to allocate and sequence the targets (or tasks), and assign a schedule for each robot such that

- each target is visited once by some robot,
- at most m tasks are processed by the human operators at any time instant,
- each robot returns to the depot at the end of its mission, and,
- the maximum mission time of any of the robots is minimized.

IV. Algorithm *Approx* for the TASSP

There are three steps in *Approx*. The first step modifies the travel times to also include the processing times of the tasks at the targets. This modification is crucial and will be later used in the proof of the approximation ratio (Lemma 2). The second step will implement the Frederickson et al. [9] algorithm to solve the *min-max* multiple TSP using the modified travel times. This step will find a sequence of targets for each robot to visit. Finally, a greedy heuristic will use the sequences found in the second step and produce feasible schedules for the robots that satisfy the constraints present due to the availability of human operators. The details in these steps are as follows:

1) **Travel cost modification:** For any edge (i,j) joining targets $i,j \in T$, let the modified travel time be $\hat{c}(i,j) = c(i,j) + \frac{p_i}{2} + \frac{p_j}{2}$. For any edge (i,d) joining a target i and depot d, let the modified travel time be $\hat{c}(i,d) = c(i,d) + \frac{p_i}{2}$. Note that the modified travel times also are symmetric and satisfy the triangle inequality.

2) **Target sequence assignment for each robot:** Solve the *min-max* multiple TSP with the modified travel times using the Frederickson et al.'s algorithm [9]. Let F denote the solution produced in this step, and let F_r denote the sequence of targets visited by robot r in F.

3) **Greedy heuristic implementation to produce feasible schedules:** Given $F := \{F_{r_1}, F_{r_2}, \cdots, F_{r_k}\}$, the heuristic works as follows: Starting at the depot, each robot will travel according to the sequence of targets assigned to it in F. When robot r arrives at a target $u \in F_r$, any human operator, if available, is immediately assigned to work on the task at target u with the robot r. If no human operator is available, the robot r simply waits for the next operator to be available. Ties are broken arbitrarily when a human operator becomes available and multiple robots are waiting to collaborate. After all the targets in F_r are visited and their corresponding tasks are completed, robot r returns to the depot.

V. Theoretical guarantee for *Approx*

Theorem 1. *The approximation ratio of Approx is equal to*

$$\begin{cases} \frac{5}{2} - \frac{1}{k} & \text{if } m \geq k, \\ \frac{7}{2} - \frac{1}{k} & \text{otherwise.} \end{cases}$$

The computational complexity of *Approx* depends mainly on the approximation algorithm by Frederickson et al. [9] which can be implemented in the order of n^3 steps. Therefore, *Approx* runs in polynomial time. We will prove the above theorem in the remainder of this section. We first show a key result that holds true for the sequence of targets F_r visited by robot r in *Approx*. This result (Lemma 1) will bound the mission time of any of the robots in terms of its travel cost and the processing times of all the tasks. Let $F_r := (u_{r1}, u_{r2}, \cdots, u_{rn_r})$ where u_{r1} is the first target visited by robot r, u_{r2} is the second target visited by robot r and so on. Let $Cost^{travel}(F_r)$ denote the travel time for robot r to visit the targets in F_r, *i.e.*, $Cost^{travel}(F_r) := c(d,u_{r1}) + \sum_{i=1}^{n_r-1} c(u_{r_i}, u_{r_{i+1}}) + c(u_{n_r}, d)$.

Lemma 1. *Given $F = \{F_{r_1}, F_{r_2}, \cdots, F_{r_k}\}$, let $Cost_r(F)$ denote the mission time of robot r after applying the greedy heuristic on F. Then,*

$$Cost_r(F) \leq Cost^{travel}(F_r) + (1 - \frac{1}{m}) \sum_{i \in F_r} p_i + \frac{\sum_{i \in T} p_i}{m}.$$

Proof: The total wait time for robot r after applying the greedy heuristic is given by $Cost_r^{wait}(F) := Cost_r(F) - Cost^{travel}(F_r) - \sum_{i \in F_r} p_i$. During this wait time, all the human operators were busy working on tasks not assigned to robot r. Therefore, as there are m human operators, $mCost_r^{wait}(F)$ must be at most equal to $\sum_{i \notin F_r} p_i$. Hence,

$$\begin{aligned} mCost_r^{wait}(F) &\leq \sum_{i \notin F_r} p_i \\ m(Cost_r(F) - Cost^{travel}(F_r) - \sum_{i \in F_r} p_i) &\leq \sum_{i \notin F_r} p_i \\ \Rightarrow Cost_r(F) &\leq Cost^{travel}(F_r) + \sum_{i \in F_r} p_i + \frac{\sum_{i \notin F_r} p_i}{m} \\ Cost_r(F) &\leq Cost^{travel}(F_r) + (1 - \frac{1}{m}) \sum_{i \in F_r} p_i + \frac{\sum_{i \in T} p_i}{m}. \end{aligned}$$

The next two Lemmas prove the approximation ratios of *Approx*.

Lemma 2. *Assume the number of human operators is at least equal to the number of robots, i.e., $m \geq k$. Then, Approx directly provides a feasible solution to the TASSP and has an approximation ratio of $\frac{5}{2} - \frac{1}{k}$.*

Proof: If $m \geq k$, it is possible to ensure there is no wait time for any robot independent of the sequences of targets assigned to the robots. This can be accomplished trivially by assigning each robot with a human operator while ensuring no operator is assigned to more than one robot. For this reason, there is always an optimal solution to the TASSP where the wait time of each robot is zero. Hence, if $m \geq k$, TASSP reduces to the problem where there are no scheduling constraints for the human operators and the objective is to find a sequence of targets for each robot such that each target is visited once by some robot and $\max_{r \in F_r}[Cost^{travel}(F_r) + \sum_{i \in F_r} p_i]$ is minimized. Note that the mission time for robot r can also be rewritten in terms of the modified travel times as follows:

$$\begin{aligned} &Cost^{travel}(F_r) + \sum_{i \in F_r} p_i \\ &= c(d,u_{r1}) + \sum_{i=1}^{n_r-1} c(u_{ri}, u_{r(i+1)}) + c(u_{rn_r}, d) + \sum_{i \in F_r} p_i \\ &= c(d,u_{r1}) + \frac{p_{u_{r1}}}{2} + \sum_{i=1}^{n_r-1} [c(u_{ri}, u_{r(i+1)}) + \frac{p_{u_{ri}}}{2} + \frac{p_{u_{r(i+1)}}}{2}] \\ &+ c(u_{rn_r}, d) + \frac{p_{u_{rn_r}}}{2} \\ &= \hat{c}(d,u_{r1}) + \sum_{i=1}^{n_r-1} \hat{c}(u_{ri}, u_{ri+1}) + \hat{c}(u_{rn_r}, d). \end{aligned} \quad (1)$$

Therefore, when $m \geq k$, TASSP reduces to solving a *min-max* multiple TSP using the modified times to travel between any two vertices. As a result, applying Frederickson et al.'s algorithm [9] directly provides an approximation ratio of $\frac{5}{2} - \frac{1}{k}$.

Let S be the solution produced by *Approx*, and let $Cost(S)$ denote the maximum mission time of any robot in S. Let $Cost^*_{tassp}(m)$ denote the optimal cost of the TASSP with

m human operators. In the next Lemma, we will prove that $Cost(S) \leq (\frac{7}{2} - \frac{1}{k})Cost^*_{tassp}(m)$ if $m < k$.

Lemma 3. *Assume the number of human operators is less than the number of robots, i.e., $m < k$. Given $F = \{F_{r_1}, F_{r_2}, \cdots, F_{r_k}\}$, let $Cost_r(F)$ denote the mission time of robot r for the feasible solution obtained using Approx. Then, $Cost(S) = \max_{r \in R} Cost_r(F) \leq (\frac{7}{2} - \frac{1}{k})Cost^*_{tassp}(m)$.*

Proof: Let $r^* := \arg\max_{r \in R} Cost_r(F)$. Using Lemma 1, we get,

$$\begin{aligned} Cost(S_2) &= \max_{r \in R} Cost_r(F) \\ &\leq Cost^{travel}(F_{r*}) + (1 - \frac{1}{m}) \sum_{i \in F_{r^*}} p_i + \frac{\sum_{i \in T} p_i}{m} \\ &\leq Cost^{travel}(F_{r*}) + \sum_{i \in F_{r^*}} p_i + \frac{\sum_{i \in T} p_i}{m}. \end{aligned} \tag{2}$$

As there are m human operators and the sum of the processing times of all the tasks is $\sum_{i \in T} p_i$, $\frac{\sum_{i \in T} p_i}{m}$ serves as a trivial lower bound for $Cost^*_{tassp}(m)$. Therefore,

$$\frac{\sum_{i \in T} p_i}{m} \leq Cost^*_{tassp}(m). \tag{3}$$

. Also, using Lemma 2, we get,

$$\begin{aligned} &Cost^{travel}(F_{r*}) + \sum_{i \in F_{r^*}} p_i \\ &\leq \max_{r \in R} [Cost^{travel}(F_r) + \sum_{i \in F_r} p_i] \\ &\leq (\frac{5}{2} - \frac{1}{k})Cost^*_{tassp}(m) \text{ for any } m \geq k. \end{aligned} \tag{4}$$

For a fixed k, note that for any two positive integers m_1, m_2 such that $m_1 \geq m_2$, $Cost^*_{tassp}(m_1)$ must be upper bounded by $Cost^*_{tassp}(m_2)$. This is due to the fact that any feasible solution to a TASSP with m_2 human operators can be trivially transformed into a feasible solution to a TASSP with m_1 human operators. Therefore,

$$\begin{aligned} &Cost^{travel}(F_{r*}) + \sum_{i \in F_{r^*}} p_i \\ &\leq (\frac{5}{2} - \frac{1}{k})Cost^*_{tassp}(m) \text{ for any } m \geq k \\ &\leq (\frac{5}{2} - \frac{1}{k})Cost^*_{tassp}(m) \text{ for any } m < k. \end{aligned} \tag{5}$$

Substituting the bounds from equations (3),(5) in equation (2) proves the Lemma.

The approximation ratio in Theorem 1 directly follows from Lemmas 2 and 3.

VI. Numerical results

Approx was coded in Julia [21] with the help of the NetworkX package [22], and the computations were run on Mac Pro (8-Core Intel Xeon E5 processor @3 GHz, 32 GB RAM). The positions of the targets in each problem instance are chosen independently and uniformly using the standard *rand* function in Julia from an area of size 50×50 units. The speed of each robot was assumed to be 1 unit, and therefore, the travel time between any two vertices in V was set to be equal to the Euclidean distance between the vertices. *Approx* was tested on a large number of instances to infer the bounds on the actual quality of solutions produced by *Approx*. This is accomplished by computing its *a-posteriori* guarantee, $i.e.$, for a given instance, the *a-posteriori* guarantee is defined as the ratio of the cost[2] of the feasible solution obtained by *Approx* and the optimal cost. These guarantees are generally lower than the approximation ratio which is a (worst case) theoretical bound true for any instance of the problem. The optimal cost was obtained by using a Mixed Integer Linear Programming Formulation of the TASSP given in the appendix. This formulation was implemented in Gurobi [23] and worked for small instances[3] (up to 12 vertices, 4 vehicles and 3 operators). For larger instances, we used the maximum of the lower bounds given by $L_1 := \frac{Cost^*_{tsp} + \sum_{i \in T} p_i}{k}$ (refer to Lemma 4 in the appendix), the trivial bounds $L_2 := \frac{\sum_{i \in T} p_i}{m}$ and $L_3 := \max_{i \in T}(2c(d,i) + p_i)$ as a proxy for the optimal cost.

The processing time of the task at each target in a problem instance is set to $\beta \bar{c}_t$ where $\beta > 0$ is a given parameter and c_t denotes the average travel times between any two targets in the instance. If β is closer to zero, the travel times dominate the processing times, $i.e.$, the TASSP reduces to a *min-max* multiple TSP. On other hand, if β is relatively high, the processing times dominate the travel times, $i.e.$, the TASSP reduces to a PMSP. In subsection VI-B, we also vary β to understand its effect on the performance of *Approx*.

A. Results for small instances

We generated 30 problem instances with at most $|V| = 12$ vertices (targets and the depot), $k = 4$ robots and $m = 3$ human operators. The processing time of the task at each target was chosen with $\beta = 0.5$. The cost of the solution obtained by *Approx*, the optimal cost and their corresponding computation times are shown in Table I. The approximation ratios as calculated by Theorem 1 for these instances are 3.16 ($= \frac{7}{2} - \frac{1}{3}$) and 3.25 ($= \frac{7}{2} - \frac{1}{4}$) for instances 1-20 and 21-30 respectively. As expected, the *a-posteriori* guarantees as shown in Table I were much lower than the approximation ratios. The average of the *a-posteriori* guarantees computed using the optimal costs was equal to ≈ 1.1. On the other hand, the average of the *a-posteriori* guarantees computed using the lower bounds was equal to ≈ 1.4. The average run time of *Approx* was ≈ 0.3 secs for the instances in this

[2]The maximum mission time of any robot.

[3]For some instances with 15 targets, the run time in Gurobi was in the order of days.

TABLE I

A-POSTERIORI GUARANTEE FOR SMALL INSTANCES

Instance No.	$\|V\|$	k	m	*Approx*		Optimal		Lower Bound	*A-posteriori* guarantee		Approximation ratio
				Cost	Run Time (secs)	Cost	Run Time (secs)		Using optimal cost	Using lower bound	
1	10	3	2	242.1	0.3	230.3	18.9	153.0	1.05	1.58	3.16
2	10	3	2	279.6	0.3	249.3	11.5	187.1	1.12	1.49	3.16
3	10	3	2	291.1	0.2	271.4	13.0	197.5	1.07	1.47	3.16
4	10	3	2	279.5	0.2	246.0	9.5	189.6	1.14	1.47	3.16
5	10	3	2	296.2	0.2	269.3	16.0	210.7	1.10	1.41	3.16
6	10	3	2	231.8	0.2	210.2	12.1	170.3	1.10	1.36	3.16
7	10	3	2	259.7	0.2	259.7	12.5	231.7	1.00	1.12	3.16
8	10	3	2	235.8	0.2	225.9	12.1	175.4	1.04	1.34	3.16
9	10	3	2	268.2	0.3	261.1	5.7	232.9	1.03	1.15	3.16
10	10	3	2	286.7	0.2	250.3	8.2	203.8	1.15	1.41	3.16
11	12	3	2	290.3	0.3	279.5	352.6	211.5	1.04	1.37	3.16
12	12	3	2	295.0	0.2	255.5	221.0	190.1	1.15	1.55	3.16
13	12	3	2	318.9	0.2	278.6	580.3	215.5	1.14	1.48	3.16
14	12	3	2	340.6	0.2	319.4	216.2	230.3	1.07	1.48	3.16
15	12	3	2	261.4	0.2	225.3	151.6	181.4	1.16	1.44	3.16
16	12	3	2	397.6	0.3	319.6	139.6	252.2	1.24	1.58	3.16
17	12	3	2	324.1	0.2	284.7	236.9	212.8	1.14	1.52	3.16
18	12	3	2	230.6	0.2	225.4	262.9	165.5	1.02	1.39	3.16
19	12	3	2	374.5	0.2	309.9	105.3	238.4	1.21	1.57	3.16
20	12	3	2	352.5	0.3	335.0	338.3	266.6	1.05	1.32	3.16
21	12	4	3	282.6	0.3	242.7	35.6	196.2	1.16	1.44	3.25
22	12	4	3	232.7	0.3	217.0	25.3	190.1	1.07	1.22	3.25
23	12	4	3	229.4	0.2	229.4	59.6	191.7	1.00	1.20	3.25
24	12	4	3	333.5	0.3	288.7	50.8	230.3	1.16	1.45	3.25
25	12	4	3	238.6	0.3	191.4	52.0	148.8	1.25	1.60	3.25
26	12	4	3	308.4	0.3	280.8	15.4	252.2	1.10	1.22	3.25
27	12	4	3	247.1	0.3	241.7	34.9	197.0	1.02	1.25	3.25
28	12	4	3	208.9	0.2	192.5	27.5	165.5	1.08	1.26	3.25
29	12	4	3	297.6	0.3	262.4	19.3	223.2	1.13	1.33	3.25
30	12	4	3	309.8	0.2	291.0	23.2	266.6	1.06	1.16	3.25

TABLE II

EFFECT OF CHANGING THE NUMBER OF HUMAN OPERATORS (m) ON THE PERFORMANCE OF $Approx$ WITH $|V|$=50, k=5 AND β=0.5

m	*Aposteriori* guarantees			Run times (secs)		
	Max	Mean	Std. deviation	Max	Mean	Std. deviation
1	1.11	1.05	0.02	0.44	0.38	0.02
2	1.27	1.16	0.04	0.45	0.38	0.02
3	1.45	1.33	0.06	0.44	0.38	0.02
4	1.60	1.40	0.07	0.44	0.38	0.02
5	1.53	1.37	0.07	0.45	0.38	0.02

TABLE III

EFFECT OF CHANGING THE NUMBER OF ROBOTS (k) ON THE PERFORMANCE OF $Approx$ WITH $|V| = 50$, $m = 1$ AND $\beta = 0.5$

k	*Aposteriori* guarantees			Run times (secs)		
	Max	Mean	Std. deviation	Max	Mean	Std. deviation
1	1.07	1.01	0.02	0.46	0.39	0.03
2	1.14	1.07	0.02	0.45	0.38	0.02
3	1.11	1.05	0.02	0.45	0.39	0.02
4	1.12	1.05	0.02	0.45	0.38	0.02
5	1.11	1.05	0.02	0.45	0.39	0.02

TABLE IV

EFFECT OF CHANGING THE NUMBER OF VERTICES ($|V|$) ON THE PERFORMANCE OF $Approx$ WITH $k = 3$, $m = 2$ AND $\beta = 0.5$

$\|V\|$	*Aposteriori* guarantees			Run times (secs)		
	Max	Mean	Std. deviation	Max	Mean	Std. deviation
10	1.74	1.42	0.15	0.25	0.24	0.00
20	1.53	1.39	0.09	0.28	0.26	0.01
30	1.54	1.36	0.06	0.30	0.28	0.01
40	1.44	1.33	0.05	0.37	0.33	0.02
50	1.33	1.27	0.03	0.45	0.39	0.02

set while the average run time of the formulation in Gurobi varied significantly with a mean value ≈ 106.75 secs.

B. Results for large instances

We generated 1200 instances to test the performance of *Approx* by systematically varying all the parameters k, m, $|V|$ and β. As we could not compute optimal solutions for these instances, *a-posteriori* guarantees were computed using the lower bounds. We implemented *Approx* on all these instances and present here the mean, maximum and standard deviation of the *a-posteriori* guarantees for the obtained solutions, and the computation times. Specifically, the following are the different sets of instances and the corresponding results:

- *Instances generated by varying the number of human operators (m)*: 100 problem instances were generated with $|V| = 50$, $k = 5$ and $\beta = 0.5$. The value of m was then varied from 1 to 5. Since $k = 5$, the approximation ratio for each instance in this set is $\frac{7}{2} - \frac{1}{5} \approx 3.3$. The results for these instances are shown in Table II.
- *Instances generated by varying the number of robots (k)*: 100 problem instances were generated with $|V| =$

TABLE V

EFFECT OF CHANGING THE PROCESSING TIMES ON THE PERFORMANCE OF *Approx* WITH $|V| = 50$, $k = 3$ AND $m = 2$

β	*Aposteriori* guarantees			Run times (secs)		
	Max	Mean	Std. deviation	Max	Mean	Std. deviation
0.1	1.66	1.44	0.10	0.44	0.36	0.02
0.5	1.44	1.28	0.05	0.56	0.40	0.04
0.9	1.22	1.14	0.03	0.55	0.44	0.04
1.3	1.15	1.10	0.02	0.61	0.47	0.04
1.7	1.14	1.09	0.02	0.98	0.56	0.11

50, $m = 1$ and $\beta = 0.5$. The value of k was then varied from 1 to 5. The approximation ratios (applying Theorem 1) for these instances are 1.5 ($k = 1$), 2 ($k = 2$), 3.16 ($k = 3$), 3.25 ($k = 4$) and 3.3 ($k = 5$). The results for these instances are shown in Table III.

- *Instances generated by varying the number of vertices ($|V|$)*: With $k = 3$, $m = 2$ and $\beta = 0.5$, we generate 100 instances for each value of $|V|$ varying from 10 to 50 in increments of 10. Since $k = 3$, the approximation ratio for each instance in this set is $\frac{7}{2} - \frac{1}{3} \approx 3.16$. The results for these instances are shown in Table IV.
- *Instances generated by varying the processing times of the tasks*: With $k = 3$, $m = 2$ and $|V| = 50$, we generate 100 instances for each value of β varying from 0.1 to 1.7 in increments of 0.4. Since $k = 3$, the approximation ratio for all these instances is again ≈ 3.16. The results for these instances are shown in Table V.

The following are the key observations from all the presented results:

1) *Approx* required less than a second to compute a feasible solution for each instance tested.
2) Varying any of the parameters ($|V|$, k, m, β) had no significant effect on the computational performance of $Approx$. This is very useful because the computation times of exact algorithms using solvers such as the Gurobi can be very uncertain and large even for instances with $|V| = 12$ (Table I).
3) As expected the *a-posteriori* guarantees were significantly better than the approximation ratio. This implies that *Approx* can find good solutions with bounds for the TASSP relatively fast.
4) As the processing times dominate the travel times (as β increases, Table V), *Approx* quickly produces solutions with improved *a-posteriori* guarantees for the TASSP.
5) In general, it was difficult to infer strong trends in the guarantees due to the following reason: If the *a-posteriori guarantee* is large for an instance, it is not clear if this is due to the poor performance of *Approx* or the lower bounding algorithms. Answering this question is outside the scope of this article and can be a topic for future research.

VII. CONCLUSIONS

This article considered a task allocation, sequencing and scheduling problem for a team of human operators and robots. An approximation algorithm which combines ideas from vehicle routing and scheduling theory was presented. Computational results were also presented to corroborate the performance of the approximation algorithm. Future work can focus on developing exact algorithms and better lower bounds. Algorithms addressing uncertainties with respect to task processing times or travel times for the robots will also be useful in practical applications.

REFERENCES

[1] M. Cummings, A. Clare, and C. Hart, "The role of human-automation consensus in multiple unmanned vehicle scheduling," *Human Factors*, vol. 52, no. 1, pp. 17–27, 2010.

[2] C. C. Murray and W. Park, "Incorporating human factor considerations in unmanned aerial vehicle routing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 860–874, 2013.

[3] L. Bertuccelli, W. Beckers, and M. Cummings, "Developing operator models for uav search scheduling," in *AIAA Guidance, Navigation, and Control Conference*, 2012.

[4] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458.

[5] H. Cai and Y. Mostofi, "A human-robot collaborative traveling salesman problem: Robotic site inspection with human assistance," in *2016 American Control Conference (ACC)*, 2016, pp. 6170–6176.

[6] T. Duckett, S. Pearson, S. Blackmore, and B. Grieve, "Agricultural robotics: The future of robotic agriculture," *CoRR*, vol. abs/1806.06762, 2018. [Online]. Available: http://arxiv.org/abs/1806.06762

[7] D. Levy, K. Sundar, and S. Rathinam, "Heuristics for routing heterogeneous unmanned vehicles with fuel constraints," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[8] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 287–294, 2014.

[9] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *17th Annual Symposium on Foundations of Computer Science (sfcs 1976)*, Oct 1976, pp. 216–227.

[10] S. Raff, "Routing and scheduling of vehicles and crews: The state of the art," *Computers & Operations Research*, vol. 10, no. 2, pp. 63 – 211, 1983, routing and Scheduling of Vehicles and Crews. The State of the Art.

[11] J. A. Castillo-Salazar, D. Landa-Silva, and R. Qu, "Workforce scheduling and routing problems: literature survey and computational study," *Annals of Operations Research*, vol. 239, no. 1, pp. 39–67, Apr 2016.

[12] S. Ponda, H.-L. Choi, and J. How, *Predictive Planning for Heterogeneous Human-Robot Teams*.

[13] M. C. Gombolay, R. J. Wilcox, and J. A. Shah, "Fast scheduling of robot teams performing tasks with temporospatial constraints," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 220–239, 2018.

[14] J. R. Peters, A. Surana, and F. Bullo, "Robust scheduling and routing for collaborative human/unmanned aerial vehicle surveillance missions," *Journal of Aerospace Information Systems*, vol. 15, no. 10, pp. 585–603, 2018.

[15] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," *Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA*, 1976.

[16] A. Frieze, "An extension of christofides heuristic to the k-person travelling salesman problem," *Discrete Applied Mathematics*, vol. 6, no. 1, pp. 79 – 83, 1983.

[17] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions Automation Science and Engineering*, vol. 4, pp. 98–104, 2007.

[18] W. Malik, S. Rathinam, and S. Darbha, "An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem," *Operations Research Letters*, vol. 35, pp. 747–753, 2007.
[19] R. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.
[20] M. Kunde, "Nonpreemptive lp-scheduling on homogeneous multiprocessor systems," *SIAM J. Comput.*, vol. 10, pp. 151–173, 1981.
[21] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.
[22] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.
[23] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2019. [Online]. Available: http://www.gurobi.com
[24] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

VIII. Appendix

A. Lower Bound

Lemma 4. *Let $Cost^*_{tassp}(m)$ denote the optimal cost of the TASSP with m human operators and let $Cost^*_{tsp}$ denote the cost of an optimal single TSP tour visiting all the vertices in V. Then, $\frac{Cost^*_{tsp}}{k} + \frac{\sum_{i\in T} p_i}{k} \leq Cost^*_{tassp}$.*

Proof: Let F^*_r be the sequence of targets visited by robot r in an optimal solution for the TASSP. Let the corresponding travel time for visiting the targets in F^*_r be denoted by $Cost^{travel*}_r$. We have, $Cost^{travel*}_r + \sum_{i\in F^*_r} p_i \leq Cost^*_{tassp}(m)$. Summing over all the k robots, we get the following inequality:

$$\sum_{r\in R} Cost^{travel*}_r + \sum_{i\in T} p_i \leq kCost^*_{tassp}(m).$$

Now, consider a feasible solution for the single TSP where a robot visits all the targets in the sequence given by $(F^*_{r_1}, F^*_{r_2}, \cdots, F^*_{r_k})$. That is, the robot starts from the depot and visits the targets in the sequence $F^*_{r_1}$, then the targets in the sequence $F^*_{r_2}$ and so on, and finally returns to the depot. Clearly, the travel time of this (single TSP) solution must be upper bounded by $\sum_{r\in R} Cost^{travel*}_r$ as the travel times satisfy the triangle inequality. Therefore,

$$\begin{aligned}\frac{Cost^*_{tsp}}{k} + \frac{\sum_{i\in T} p_i}{k} &\leq \frac{\sum_{r\in R} Cost^{travel*}_r}{k} + \frac{\sum_{i\in T} p_i}{k} \\ &\leq Cost^*_{tassp}(m).\end{aligned}$$

B. Mixed Integer Linear Program

Decision Variables: For all $i, j \in V, i \neq j$, the binary variable x_{ij} is equal to 1 if a robot is traveling from i to j and is equal to 0 otherwise. Similarly, for all $u, v \in V$, $u \neq v$, y_{uv} is equal to 1 if an operator works on the task at v immediately after the task at u, and is equal to zero otherwise. There are also continuous variables defined as follows:
vet_i = time at which a robot reaches target $i \in T$.
vlt_i = time at which a robot leaves target $i \in T$.
tst_i = time at which the processing of task at $i \in T$ is started.
MT = maximum mission time of any robot.

Objective: The objective is to minimize the maximum mission time of any robot, $i.e.$, $\min MT$.
Constraints:
Number of incoming and outgoing edges from the depot is equal to the number of robots. That is,

$$\sum_{j\in T} x_{jd} = \sum_{j\in T} x_{dj} = k. \tag{6}$$

There is exactly one incoming and outgoing travel edge for each target. That is,

$$\sum_{i\in V\setminus j} x_{ij} = \sum_{i\in V\setminus j} x_{ji} = 1 \ \forall j \in T. \tag{7}$$

The depot is considered as a dummy vertex from which the operators leave for processing the tasks and return after completing them. That is,

$$\sum_{j\in T} y_{dj} = \sum_{j\in T} y_{jd} = m. \tag{8}$$

An operator can work on at most one other task immediately after completing a task at a target. That is,

$$\sum_{i\in V\setminus j} y_{ij} = \sum_{i\in V\setminus j} y_{ji} = 1 \ \forall j \in T. \tag{9}$$

The earliest a robot can reach a target is at least equal to the sum of the completion time of its previous task (if any) and the time taken to travel from its previous vertex. That is,

$$(vet_j - tst_i)x_{ij} \geq (c(i,j) + p_i)x_{ij} \quad \forall i \in T, j \in T \setminus i. \tag{10}$$

The earliest an operator can process a task is after completing his/her prior task if any. That is,

$$(tst_j - tst_i)y_{ij} \geq p_i y_{ij} \quad \forall i \in T, j \in T \setminus i. \tag{11}$$

The earliest the robot can reach a target is at least equal to the time taken for it to travel from the depot to the target. That is,

$$vet_i \geq c(d,i)x_{di} \ \forall i \in T. \tag{12}$$

The earliest a task can be processed is at least after the robot has reached the target.

$$tst_i \geq vet_i \ \forall i \in T. \tag{13}$$

For any target, the maximum mission time, MT, must be at least equal to the sum of the task completion time at the target and the time taken to return from the target to the depot. That is,

$$tst_i + p_i + c(i,d)x_{id} \leq MT \quad \forall i \in T. \tag{14}$$

Though constraints (10)–(11) contain bi-linear terms, they can be linearized using the McCormick relaxation [24] without loosing any information, as the relaxation is exact for bi-linear variables. So, the formulation can be recast as a Mixed Integer Linear Program (MILP).