

Article

Research on Multi-Objective Multi-Robot Task Allocation by Lin–Kernighan–Helsgaun Guided Evolutionary Algorithms

Zhenqiang Zhang ¹, Sile Ma ^{1,2} and Xiangyuan Jiang ^{1,*}¹ Institute of Marine Science and Technology, Shandong University, Qingdao 266237, China² School of Control Science and Engineering, Shandong University, Jinan 250061, China

* Correspondence: xyjiang@sdu.edu.cn

Abstract: Multi-robot task allocation (MRTA) and route planning are crucial for a large-scale multi-robot system. In this paper, the problem is formulated to minimize the total energy consumption and overall task completion time simultaneously, with some constraints taken into consideration. To represent a solution, a novel one-chromosome representation technique is proposed, which eases the consequent genetic operations and the construction of the cost matrix. Lin–Kernighan–Helsgaun (LKH), a highly efficient sub-tour planner, is employed to generate prophet generation beforehand as well as guide the evolutionary direction during the proceeding of multi-objective evolutionary algorithms, aiming to promote convergence of the Pareto front. Numerical experiments on the benchmark show the LKH guidance mechanism is effective for two famous multi-objective evolutionary algorithms, namely multi-objective evolutionary algorithm based on decomposition (MOEA/D) and non-dominated sorting genetic algorithm (NSGA), of which LKH-guided NSGA exhibits the best performance on three predefined indicators, namely C-metric, HV, and Spacing, respectively. The generalization experiment on a multiple depots MRTA problem with constraints further demonstrates the effectiveness of the proposed approach for practical decision making.



Citation: Zhang, Z.; Ma, S.; Jiang, X. Research on Multi-Objective Multi-Robot Task Allocation by Lin–Kernighan–Helsgaun Guided Evolutionary Algorithms. *Mathematics* **2022**, *10*, 4714. <https://doi.org/10.3390/math10244714>

Academic Editor: Daniel-Ioan Curiac

Received: 16 November 2022

Accepted: 7 December 2022

Published: 12 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: multi-robot task allocation; route planning; multi-objective evolutionary algorithm; Lin–Kernighan–Helsgaun; non-dominated sorting genetic algorithm; Pareto front; decision making

MSC: 90C27

1. Introduction

Today with the emphasis on environmental sustainability and digital transformation [1,2], people are looking for proper measures to progress green manufacturing [3,4], urban sustainability, etc. As a potential paradigm to enable this progress, the multi-robot system (MRS), or swarm robotics, has attracted considerable attention in the last two decades [5–7]. An MRS is a swarm of robots that are designed to commit large-scale collective tasks that are difficult or impossible to address by a single robot, thus decreasing the overall make-span time as well as promoting the efficiency of tasks completion. With MRS, the system's flexibility and overall robustness can be assured, and the individual robot's design can be simplified, too [8]. MRS are playing an important role in the context of Industry 4.0, applications include but are not limited to modern logistics [9], surveillance [10], intelligent manufacturing [11], wireless sensor networks [12], etc.

For most MRS cooperative task scenarios, tasks can be viewed as spatially located points identified by their coordinates that need to be visited by the robots in the group; thus, a fundamental and important problem arises to allocate whom to which task and in what order for each robot to complete its assigned tasks, so that the overall system performance can be maximized. This problem is known as MRTA (multi-robot task allocation) [6,13,14]. MRTA problems can be classified as follows as per the entities or requirements of MRS [14]:

- Single-task robots (ST) or multi-task robots (MT), according to single robot's capability of how many tasks it can execute at a time.

- Single-robot tasks (SR) or multi-robot tasks (MR), according to the task's requirement of number of robots to fulfill it.
- Instantaneous assignment (IA) or time-extend assignment (TA), depending on the timeliness of the solution. IA means the solution is instantaneous but may be short-sighted, while TA uses more information of the situation and thus can achieve a plan of global sense.

Generally, the MRTA approaches can be classified into two main categories according to the organizational paradigm of robots [13,15,16]: centralized and decentralized. For the centralized approaches, there is a central planner (e.g., control station) who uses the global knowledge of the system to produce optimal or sub-optimal solutions, while for the decentralized approaches, the solution is drawn by all or part of the robots together, in a distributed and cooperative fashion, using only the local information available.

This paper focus on the ST-SR-TA problem, which is more common in MRTA. See Figure 1 for example: how to assign a swarm of UAVs, i.e., $R_1 \dots R_3$ in this case, to survey all the targets located in the urban area, i.e., $T_1 \dots T_{12}$, so that the global efficiency can be maximized? In these type of cases: (a) each robot can only take at most one task at a moment, (b) each task needs one robot to accomplish it, and (c) the time to deduce the solution is not necessarily instantaneous. The centralized approaches will be studied herein because in most cases, before launching the swarm, the decision maker wants to know the assignment beforehand and modify it if possible.

Some papers used market-based approaches to tackle the problem, which mimic the human society market economies mechanism to maximize the individual agent's profit and at the same time achieve a collective profit [17,18]. In these approaches, the robots bid on tasks under auctioning, either in sequences or clusters, according to their evaluation of each task. The solution quality, however, cannot be guaranteed because of the intrinsic greedy heuristics of the market-based algorithms. The mostly researched approaches are the heuristic methods, which search in the solution space for near-optimal solutions that satisfy some certain objectives globally. These methods include particle swarm optimization (PSO) [19], genetic algorithm (GA) [20], Tabu-list search (TS) [21], etc., which are customized to MRTA problems. Due to the exploration and exploitation nature of the algorithms, heuristic-based methods often deduce satisfactory solutions.

Actually, for practical MRTA, we expect the swarm to accomplish the tasks as soon as possible while keeping the cost at a low level, possibly with some constraints in some scenarios. Thus, in this paper, we care about minimizing two objectives simultaneously: (a) the overall completion time and (b) the total cost. Unfortunately, the two are optimized in a contradictory way: for example, the lowest total cost might mean that one robot takes all the tasks, leading to a high overall completion time due to the lack of cooperation among robots. The best trade-offs among the multi-objectives are defined in terms of Pareto optimality [22], and the set of all the Pareto optimal solutions form the called Pareto set. The Pareto set is extremely valuable to the decision making of multi-objective MRTA; thus, this paper aims to find effective methods to resolve the Pareto set.

The contribution of the paper is as follows:

- To code a feasible solution concisely, a novel one-chromosome representation technique is proposed. Compared to its traditional counterparts such as two-part chromosome, it first deems the depots of robots as virtual tasks in MRTA, thus making genetic operations easier and ensuring the diversity of the population.
- The Lin-Kernighan-Helsgaun (LKH) guidance mechanism, as an efficient local planner, is introduced into the multi-objective evolutionary algorithms (MOEA) in order to generate prophet generation as well as guide the evolutionary direction during the evolutionary process, which is proven effective to improve the solution quality of the Pareto set. Moreover, the guidance mechanism is specifically designed to take effect in a probabilistic sense so that it would not lead to additional computational burden.
- The guidance rate as the main parameter of the LKH guidance mechanism has been studied, and the proper value is given.

- Numerical experiments show the algorithm is helpful in providing near optimal solutions for practical decision making in MRTA with constraints fulfilled.

The remainder of the paper is organized as follows: Section 2 briefly reviews the works related to the multi-objective optimization for MRTA. Section 3 models the problem in mathematical formulation, with bi-objective and two constraints. In Section 4, we propose the LKH guided multi-objective evolutionary algorithm, which implements a novel one-chromosome representation technique to code a solution and uses the LKH guidance mechanism to guide the evolutionary direction of MOEA; Section 5 arranges the numerical experiments and compares five different approaches to demonstrate the effectiveness of the proposed methods. In Section 6, the proposed method is used to solve a more general MRTA problem—multi-depot MRTA with constraints, showing how our approach can help practical decision making. Finally, Section 7 concludes the paper.

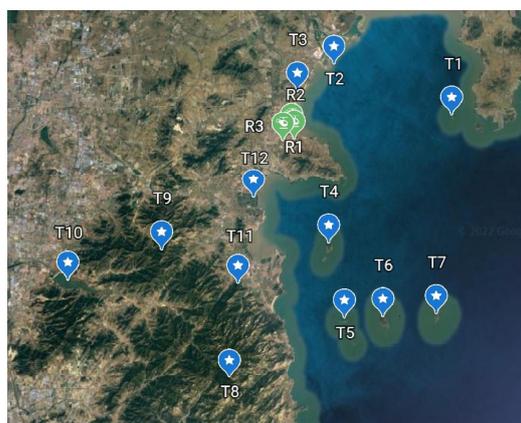


Figure 1. A typical MRTA problem scenario: three robots (green) to survey fourteen tasks (blue-stars).

2. Related Work

As market-base approaches are based on greedy strategy and usually fail to give out optimal solutions to MRTA problems, we only review heuristic methods herein. Various works attribute MRTA problem to MTSP (multiple salesman problem) [13,15,23]. As this is a combinatorial optimization problem (COP), the imperative step is how to code a solution. Ref. [24] proposed the famous two-part chromosome representation, which has a clear structure. However, the authors in [25] figure out there is a limitation on the diversity of the genetic operation of the two-part chromosome, and they proposed a new crossover approach named TCX to overcome this advantage. In addition, multi-chromosome is another alternative representation technique, as proposed and used in [26,27]. Both the techniques suffer from complex genetic operations because the chromosome must be split first so that the operations can be executed. A more concise representation technique is preferred.

As for the heuristic optimization methods for MRTA, Ref. [20] developed a combinatorial algorithm in the context of an industrial plant inspection scenario, using GA for task allocation and the A* algorithm for collision-free path planning, respectively. However, the two objectives in their work—minimization of total fuel consumption and minimization of task completion time—are resolved separately. Similar to this, the work in [28] used GA and LKH to iteratively solve the tasks assignment problem in the context of car door spot welding robots. The objective is established based on the balance of the welding tasks of each robot, aiming to result in less welding time and awaiting time. In [19], a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is proposed, with the objective to minimize both the overall team cost and any individual robot's workload simultaneously. The algorithm extended the standard single-objective PSO to cope with discrete spaces and multiple objectives. The result obtained a well-balanced workload among robots; however, no proper evaluation indicator is presented. The authors in [29] presented a multi-objective approach in designing a decision support system for underwater cleaning robots, using a

Probabilistic Roadmap (PRM) to explore all the feasible paths and Non-dominated Sorting Genetic Algorithm (NSGA) to optimize the sequential route for each robot. The objectives including the path length and routing angle are considered to be optimized, while ensuring constraints such as maximum travel distance and cable entanglement. Additionally, the NSGA-based algorithms are compared with MOPSO and show that the NSGA-III produced the best solution quality. A novel framework for multi-UAV task scheduling has been proposed in [21], including two iterative phases: the first is the task allocation phase and the second is the single UAV scheduling phase. Tabu-list-based simulated annealing (SATL) and variable neighborhood descent (VND) algorithms are used to solve the two phases separately. The objective, however, is formulated to maximize the total profit of scheduled tasks only.

As discussed, the previous works mainly focus on single-objective MRTA. Not so many are concerned with the multi-objective counterpart, which is our concern. Actually, both mathematical programming approaches and meta-heuristics are available for multi-objective optimization problems [30]. The mathematical programming approaches such as multi-objective branch and bound (MOB&B) [31] and Benson’s algorithm for multi-objective linear programming [32], however, have limitations that are only adaptive to specific features of the problem to be solved, or they confront the problem of a dimension curse. In contrast, the multi-objective evolutionary algorithms (MOEA) are more popular as they are less susceptible to the shape and continuity of the Pareto front, and they are also capable of finding members of a Pareto set in a single run [30]. Such approaches include the strength Pareto evolutionary algorithm (SPEA) [33], non-dominated genetic algorithm (NSGA) [34,35], S-metric selection evolutionary multi-objective algorithm (SMS-EMOA) [36], directed search domain (DSD) [37], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [22], etc.

There is no doubt that the variety of MOEAs will offer various methods to tackle with the multi-objective MRTA problem in this paper. Yet we believe some extra development should be made in order to adapt the problem and promote the performance further.

3. Problem Formulation

We formulate the ST-SR-TA MRTA problem as a combinational optimization problem (COP), the solution being proven NP-hard [15,23,26]. We will not distinguish between single-depot or multi-depot MRTA because the former is a special case of the latter. We will only consider closed-loop MRTA herein, where each robot returns to its depot after visiting the tasks assigned to it.

Let $R = \{R_1, \dots, R_n\}$ be the set of robots and $T = \{T_1, \dots, T_m\}$ be the set of tasks. Let $\Omega_k \subseteq T, k \in \{1, \dots, n\}$ be a sequence of tasks assigned to robot k and let $q(\Omega_k)$ be its cost. Let $x_{kij}, i \neq j$ be a binary decision variable that indicates whether or not robot k travels from task point i to j , $x_{kij} = 1$ if yes; otherwise, $x_{kij} = 0$. The objective of the task allocation is to find an allocation $\{\Omega\}^*$ that minimizes the total cost of the group, denoted as F_1 , and the maximum cost of each robot, denoted as F_2 , simultaneously:

$$\{\Omega\}^* = \operatorname{argmin}(F_1, F_2) \tag{1}$$

$$F_1 = \sum_{k=1}^n q(\Omega_k) \tag{2}$$

$$F_2 = \max\{q(\Omega_k)\} \tag{3}$$

$$q(\Omega_k) = \sum_{i=1}^m \sum_{j=1}^m x_{kij}c_{kij}, \forall k \in \{1, \dots, n\} \tag{4}$$

s.t.

$$\sum_{k=1}^n \sum_{j=1}^m x_{kij} = 1, \forall i \in \{1, \dots, m\} \tag{5}$$

$$\sum_{k=1}^n \sum_{i=1}^m x_{kij} = 1, \forall j \in \{1, \dots, m\} \tag{6}$$

$$\sum_{i,j \in S} x_{kij} \leq |S| - 1, \forall S \subset \Omega_k \setminus \{1\}, \forall k \in \{1, \dots, n\} \tag{7}$$

$$x_{kij} = \begin{cases} 1, & \text{if robot } k \text{ travels from } i \text{ to } j, i \neq j; \\ 0, & \text{otherwise.} \end{cases}, \forall k \in \{1, \dots, n\} \tag{8}$$

where Equations (5) and (6) ensure one task is being asked by the group only once, and in Equation (7) prevents an unexpected sub-loop inside robot k 's tour, which is also named as the sub-tour elimination constraints (SECs) [38].

The scalar c_{kij} in (4) is the cost for robot k traveling from i to j . It can be calculated using the energy consumption, or the traveling distance directly, taking the avoidance of static obstacles in the environment into consideration if necessary. For convenience, we use the Euclid distance in the following sections. For centralized MRTA, all the costs can be calculated in advance for ease of usage as a cost matrix:

$$A_k = \begin{bmatrix} c_{k11} & \dots & c_{k1n} \\ \vdots & \ddots & \vdots \\ c_{kn1} & \dots & c_{knn} \end{bmatrix} \tag{9}$$

where the elements on the diagonal are all equal to 0 (neglecting the cost on the task itself). As for homogeneous robots in this paper, A_k is the same for every robot; thus, A_k is simplified as A .

Usually, for practical applications, the robot has limited energy to ensure a long enough travel; hence, the mileage constraint needs to be taken into consideration. For some time-sensitive tasks, for example a chemical plant to be surveyed may only operate in a certain period, so the time-window needs to be considered, too. Hence, there are two more constraints:

$$q(\Omega_k) \leq L_k, \forall k \in \{1, \dots, n\} \tag{10}$$

$$TW_i^S \leq t_{ki} \leq TW_i^E, \text{ if } i \subseteq \Omega_k, \forall i \in \{1, \dots, m\} \tag{11}$$

where L_k is the mileage ability of robot k , and t_{ki} is the moment it reaches task i . The start and end time of the time-window of task i are TW_i^S and TW_i^E respectively, with $TW_i^S < TW_i^E$. The constraints in (10) and (11) are to be handled using penalty functions, that is to say, add the below item to the objective function:

$$\overline{P(\Omega)} = \alpha \times \sum_{k=1}^n \max\{0, q(\Omega_k) - L_k\} + \beta \times \sum_{i=1}^m \max\{0, TW_i^S - t_i\} + \gamma \times \sum_{i=1}^m \max\{0, t_i - TW_i^E\} \tag{12}$$

where α, β, γ are three positive adjustable penalty factors implying the significance of the corresponding violation is to the problem. Should one constraint be more strictly concerned, its corresponding penalty factor must be set relatively bigger. Of course, a big enough value, such as $+\infty$, means no violation is allowed for the constraint.

Note the objective function (3) aims to minimize the maximum cost of each robot in the group; thus, a minimum overall completion time can be attained. It can also be alternated as the amplitude of the costs among robots in some related works [39,40], which ensures a balanced workload among robots, i.e.:

$$F_2 = \max\{q(\Omega_k)\} - \min\{q(\Omega_k)\} \tag{13}$$

For convenience, we name the objective functions of (2), (3) and (13) as *minsum*, *minmax*, and *minamp*, respectively. Unlike its single-objective counterpart, the optimization of multiple objectives is more complex, because the minimization of one objective may worsen another objective. Actually, the solution of multiple objective optimization (MOP)

is not unique but a solution set, any element in the set is a feasible solution, and we cannot say a solution is better than another without specifying the criterion to be compared. In the following section, the solution set of MOP is solved by multi-objective evolutionary algorithms, with some enhancements that are proposed in this paper.

4. Methodology

Multi-objective evolutionary algorithms (MOEA), as discussed in Section 2, are the most popular methods to solve MOP. In this section, we will adapt and enhance them to solve the MRTA problem. Firstly, a one-chromosome representation technique is designed to code the solution. With the concise representation, normal genetic operations can easily be proceeded, and the formulated cost matrix is intuitive. Then, the LKH guidance mechanism will be introduced to the conventional MOEAs in order to promote convergence. Two typical varieties of MOEA, namely NSGA and MOEA/D, are studied therein.

4.1. Solution Coding

The most commonly used representation techniques for MRTA are two-chromosome, two-part chromosome [24], and multi-chromosome [26], of which the two-part chromosome representation is the most preferred. Figure 2a is a showcase of this representation, where robot R_1 takes tasks T_7, T_5, T_2 , robot R_2 takes tasks T_3, T_6 , and robot R_3 takes $T_4, T_8, T_9, T_{10}, T_1$. The limitation of the two-part chromosome technique lies in the genetic operation on the second part of the chromosome. Take Figure 2b for example: the tasks per robot would be limited to certain numbers; thus, the population diversity would be limited. In [25], a new crossover approach called TCX (two-part chromosome crossover) is proposed to overcome this drawback, but the genetic operations are complicated.

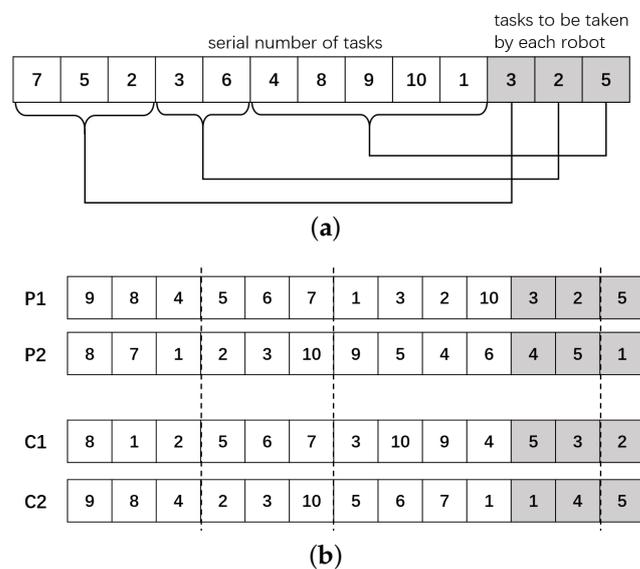


Figure 2. Example of (a) two-part chromosome representation and (b) crossover operation.

In this paper, we design a new one-chromosome representation. The reason we resort to one-chromosome representation is that the genetic operations are easy to implement, and the population diversity can be realized using normal genetic operations. An example for 10 tasks and 4 robots is shown in Figure 3. We combine the depots of each robot and the tasks as a whole by denoting the codes of depots as consequences after the last task, such as no. 11, 12, 13, and 14 in Figure 3a. In appearance, it is a standard traveling salesman problem (TSP) representation, so normal genetic operations can be implemented, but we define a different meaning for this representation: the robot from one depot would visit its following tasks numbers until the next depot is met. Take robot R_1 in Figure 3a for example, its depot is denoted as virtual no. 11, and it will visit tasks T_7 and T_5 in sequence and then

return to its depot. Virtual no. 13 is the depot of robot R_3 , and it will take T_2, T_3, T_6 as its tasks, and so on.

To simplify the calculation of the cost function, we extend the cost matrix A to A_e , as is shown in Figure 3b, where the extra elements, i.e., costs between any of the depots and tasks, should be calculated in advance. This is worthwhile for a centralized MRTA problem because it is carried out only once and can be used as a lookup table in the succeeding processes.

| | | | | | | | | | | | | | |
|----|---|---|----|---|---|---|----|---|---|---|----|----|---|
| 11 | 7 | 5 | 13 | 2 | 3 | 6 | 12 | 4 | 8 | 9 | 14 | 10 | 1 |
|----|---|---|----|---|---|---|----|---|---|---|----|----|---|

(a)

$$A_e =$$

| | | | | | |
|------------|---------|-------------|-------------|-------------|-------------|
| A | | $c_{1,11}$ | $c_{1,12}$ | $c_{1,13}$ | $c_{1,14}$ |
| | | \vdots | \vdots | \vdots | \vdots |
| $c_{11,1}$ | \dots | 0 | $c_{11,12}$ | $c_{11,13}$ | $c_{11,14}$ |
| $c_{12,1}$ | \dots | $c_{12,11}$ | 0 | $c_{12,13}$ | $c_{12,14}$ |
| $c_{13,1}$ | \dots | $c_{13,11}$ | $c_{13,12}$ | 0 | $c_{13,14}$ |
| $c_{14,1}$ | \dots | $c_{14,11}$ | $c_{14,12}$ | $c_{14,13}$ | 0 |

(b)

Figure 3. Example of (a) the proposed one chromosome representation and (b) the extended cost function.

4.2. Genetic Operations

With the proposed one-chromosome representation, normal genetic operations can be proceeded, including selection, crossover, mutation, and re-insertion. For crossover and mutation operations, we use PMX (Partial Matched Crossover) [41] and inversion, respectively. For selection operation, we use RWS (roulette wheel selection) to select the elites for offspring generation. Note for a multi-objective problem, we not only use objective functions but also non-dominate comparison and crowding distance for fitness evaluation [34]. Re-insertion is a special operation in NSGA, where the offspring are re-inserted into the father population; then, we use fitness evaluation for elites reservation.

4.3. LKH Guidance

The multi-objective evolutionary algorithms, such as NSGA and MOEA/D, are effective in finding multi-objective solutions. As mentioned previously, the solution is not unique, but a set of them constitutes the Pareto set. In practical decision-making systems, we expect the algorithm to converge quickly while achieving high-quality solutions at the same time. Actually, for MRTA, we can divide the problem into two hierarchical processes: task allocation and sub-tours costs optimization. The latter is indeed a TSP problem to optimize the visiting sequence of the nodes in the sub-tour. So, we need a proper solver to tackle TSP efficiently. Exact solvers use Integer Linear Programming (ILP) to solve the problem, but the intrinsic computational complexity renders them impractical for large-scale problems. In contrast, heuristic solvers search for near-optimal solutions with much lower complexity and thus are more desirable for real-life applications where statistically good performance is the goal. Of all the heuristic TSP solvers, Lin–Kernighan–Helsgaun (LKH) [42] is generally considered a powerful algorithm at present.

LKH is a local optimization algorithm developed based on λ -opt move, where λ edges in the current tour are replaced by another set of candidate λ edges to achieve a shorter tour. Starting from a randomly initialized tour, it iteratively searches for λ -opt exchanges that improve the tour, using a depth-first strategy. LKH has been proven to produce optimal solutions for almost all problems with a known optimum [42]. Although the running time is approximately $O(n^{2.2})$, the efficiency is satisfactory for practical industrial problems.

In our work we will firstly use LKH as an initializer to generate the prophet population; then, we will use LKH to guide the evolutionary direction during the process of a multi-objective evolutionary algorithm, resulting in Algorithm 1 (NSGA for example). The input to the algorithm is the tasks points' coordinate $\{X_i\}$ and robots depots' coordinate $\{X_k\}$, and the output is the Pareto set, which constitutes a set of assignments of each robot's tasks and the corresponding visiting sequence $\{\Omega_k\}$. The extended cost matrix A_e in line 1 contains not only the cost information among the tasks but also the cost information from the depots to the tasks. Lines 3–6 intend to generate the prophet population, where each individual's sub-tours are optimized by LKH. In lines 9–18, the LKH algorithm tries to optimize each individual of the population with probability p_{LKH} . Note that a higher p_{LKH} does not imperatively improve the solution quality, because it benefits very little at the end of the iteration, but it may consume much computational time. See Section 5.3 for a more detailed discussion. The $q(\Omega_k)$ in lines 4 and 11 means the cost of the robot k 's sub-tour, which can be derived by summing up the looked-up values from A_e . The $ObjV_1$ and $ObjV_2$, i.e., sum of and max of $\{q(\Omega_k)\}$, respectively, are crucial to the algorithm, they form the so-called Pareto front which is the concrete optimization target of the algorithm.

Algorithm 1: LKH-guided NSGA for MRTA problem

Input: tasks points' coordinate $\{X_i\}$; depots' coordinate $\{X_k\}$; LKH guidance probability p_{LKH} ;
Output: assigned each robot's tasks and corresponding visiting sequence $\{\Omega_k\}$;

- 1 Use $\{X_i\}$ and $\{X_k\}$ to calculate the extended cost matrix A_e ;
- 2 Randomly initialize NSGA populations $\{P_{i_pop}\}$;
- 3 **for** each individual in $\{P_{i_pop}\}$ **do**
- 4 $\{\Omega_k\}_{i_pop}, \{q(\Omega_k)\}_{i_pop} \leftarrow$ use LKH to optimize the sub-tours;
- 5 $[ObjV_1, ObjV_2]_{i_pop} \leftarrow [sum(\{q(\Omega_k)\}_{i_pop}), max(\{q(\Omega_k)\}_{i_pop})]$;
- 6 **end**
- 7 **while** not termination criteria **do**
- 8 selection, crossover, mutation operations to generate offspring;
- 9 **if** $rand(0, 1) < p_{LKH}$ **then**
- 10 **for** each individual in $\{P_{i_pop}\}$ **do**
- 11 $\{\Omega_k\}'_{i_pop}, \{q(\Omega_k)\}'_{i_pop} \leftarrow$ try to use LKH to optimize the sub-tours;
- 12 $[ObjV_1, ObjV_2]'_{i_pop} \leftarrow [sum(\{q(\Omega_k)\}'_{i_pop}), max(\{q(\Omega_k)\}'_{i_pop})]$;
- 13 **if** $[ObjV_1, ObjV_2]'_{i_pop} \prec [ObjV_1, ObjV_2]_{i_pop}$ **then**
- 14 $\{\Omega_k\}_{i_pop} \leftarrow \{\Omega_k\}'_{i_pop}$;
- 15 $[ObjV_1, ObjV_2]_{i_pop} \leftarrow [ObjV_1, ObjV_2]'_{i_pop}$
- 16 **end**
- 17 **end**
- 18 **end**
- 19 insert the offspring to the population;
- 20 non-dominate sorting, crowding distance evaluation to reserve the elites only;
- 21 **end**

5. Numerical Experiments

As lacking benchmarks of general MRTA problems, we use single-depot MTSP (SD-MTSP) dataset for comparison with the existing approaches. The benchmark we used is

the mTSPLib (<https://profs.info.uaic.ro/~mtsplib/>, accessed on 6 December 2022) defined by Necula et al. [39], which includes 4 main instances, i.e., *eil51*, *berlin52*, *eil76*, and *rat99*, each instance has different number of salesmen. For example, *eil51*, $n = 5$ is an instance composed of 51 cities and 5 salesmen, where the first city is assigned as the common depot that all the salesmen should start with. Two objectives, minimization of the total travel distance and minimization of the maximum travel distance of each salesman, are to be optimized simultaneously. No constraints are deployed in this benchmark.

In this section, in order to illustrate the performance of our proposed LKH guided approach, we'd first dedicate comparison among 5 different approaches: (a) g-MinMaxACS, (b) MOEA/D, (c) NSGA, (d) MOEA/D with LKH guidance, and (e) NSGA with LKH guidance. Then we'd study the influence of guidance parameter on the performance of the approach.

5.1. Parameters Setup

g-MinMaxACS: The authors in [39] proposed and investigated several Ant Colony System (ACS) based approaches, of which the g-MinMaxACS is reported the best overall performance, achieving good trade-off solutions that are diverse and in proximity of the reference Pareto front. We use this g-MinMaxACS approach and set the parameters as: $n_{ants} = 100$, $q_0 = 0.9$, $\rho = 0.1$, $\beta = 2.0$.

MOEA/D: The algorithm is based on the decomposition idea to decompose the multi-objective optimization problem into a number of scalar optimization sub-problems and optimize them simultaneously [22]. We adapt the algorithm to solve MRTA problems, for comparability with other algorithms, the parameters are set as follows: population size $n_{pop} = 100$, rate of crossover $p_x = 0.5$, rate of mutation $p_m = 1$.

NSGA: The most popular NSGA-II (without guidance) is used herein. To make it comparable, we set the parameters to be the same as MOEA/D: population size $n_{pop} = 100$, rate of crossover $p_x = 0.5$, and rate of mutation $p_m = 1$.

MOEA/D and NSGA with LKH guidance: The LKH-guided MOEA/D as well as NSGA, as we proposed in the paper, are with the same parameters as their counterpart without guidance, but we add one more parameter: guidance rate $p_{LKH} = 0.002$, meaning that the algorithm has a chance of an average of two times guidance every 1000 iterations.

Above each algorithm, we executed for a maximum number of iterations of 1500 as the stopping criteria. All the multi-objective evolutionary algorithms are programmed in Python language with *geatpy* toolbox (<http://geatpy.com/>, accessed on 6 December 2022).

5.2. Evaluation Metrics

The solution quality of an algorithm is evaluated from three aspects, i.e., convergence, diversity, and uniformity, which are quantified by the below listed three indicators. Note that the ground true Pareto fronts are not known for the mTSPLib instances [39], so the convergence can only be evaluated by some indicators indirectly.

C-metric: We use the C-metric to indicate the convergence of the solution set indirectly, which is computed as the fraction of members of set B that are weakly dominated by any member of set A [43]:

$$C(A, B) = \frac{|\{b \in B; \exists a \in A : a \preceq b\}|}{|B|} \quad (14)$$

The function $C(A, B)$ maps the pair (A, B) in comparison into the interval $[0, 1]$. To make it simple, we define the solution set of the algorithm in evaluation as B and all the other algorithms as A . So, the lower the C-metric is, the better the solution set of B is, in other words, the more likely to converge to the true Pareto front.

HV: Hyper-volume is the most commonly used indicator to indicate both the convergence and diversity of the Pareto set. Given a solution set P and a reference point r , HV can be computed as follows:

$$HV(P) = \lambda(\cup_{p \in P} \{x | p \prec x \prec r\}) \quad (15)$$

where λ denotes the Lebesgue measure [43]. Actually, the HV value can be seen as the volume of the union of the hyper-volumes determined by each element of the solution set and the shared reference point. So, a larger HV value implies the better convergence and diversity of the solution set.

Spacing: We use the *spacing* to gauge the uniformity of the solution set, which is defined as follows [43]:

$$Spacing(P) = \sqrt{\frac{1}{|P|-1} \sum_{i=1}^{|P|-1} (\bar{d} - d_i)^2} \quad (16)$$

where d_i is the Euclidean distance between the i -th solution and its next neighbor in the objective space, and \bar{d} is the mean value of all d_i . A lower *spacing* value means the solution set is more uniform.

5.3. Result and Discussion

We perform the five algorithms on eight instances: *eil51, n = 5*; *eil51, n = 7*; *berlin52, n = 5*; *berlin52, n = 7*; *eil76, n = 5*; *eil76, n = 7*; *rat99, n = 5*; *rat99, n = 7*. The obtained Pareto fronts can be seen in Figure 4. Generally speaking, for minimization problems, objective values close to the left and bottom of the axes of the graph indicate better convergence to the true Pareto front. At the same time, the wide and even spread of the objective values means good performance on diversity and uniformity. It can be observed from Figure 4 that the LKH guidance mechanism is effective for the solutions quality improvement, both for MOEA/D and NSGA algorithms; i.e., they dominate most of the solutions of the rest of the algorithms, and they are much closer to the left and bottom side of the graph. This phenomenon is much clearer for large-scale problems such as *eil76* and *rat99*. On *rat99, n = 7*, which is the largest instance, LKH-guided algorithms gain the most obvious improvements compared with other instances, reminding that the guidance rate is only two times every 1000 iterations ($p_{LKH} = 0.002$). It can also be observed that the NSGA algorithms, with or without LKH guidance, perform better than the MOEA/D counterparts.

Even though the solution set of g-MinMaxACO can sometimes dominate some other algorithms partially, the main drawback is that they are too crowded in the objective space, lacking diversity, which puts a huge limitation on the scope of candidates that the decision maker can choose from. The possible reason is that the pheromone mechanism of ACO tends to attract the ants to select similar edges. On the contrary, the evolutionary algorithm-based approaches, no matter with or without LKH guidance, tend to produce a more diverse solution set in which the decision maker has more choices.

Detailed comparative information is shown in Table 1, each algorithm is evaluated by the three metrics designed in Section 5.2, on all the instances. The best performance is highlighted with black bold in the table. It can be observed that the LKH-guided NSGA outperforms the others on the indicators *C-metric*, *HV*, and *spacing*, except for the instance *berlin52, n = 7*. To address how much the improvement is, the relative improvements of LKH-NSGA with respect to the rest of the algorithms are given in Table 1, too. The relative improvements for *C-Metric* and *Spacing* indicators are computed as the reduced percent with respect to the minimum of the rest of the algorithms because these two indicators should be as small as possible, while for the *HV* indicator, the percent increased with respect to the maximum of the rest of the algorithms because it should be as high as possible. We can say that with the proposed LKH guidance mechanism, the solutions quality of both MOEA/D and NSGA are promoted dramatically, especially when the situation comes to large-scale problems, and the LKH-guided NSGA is more preferred.

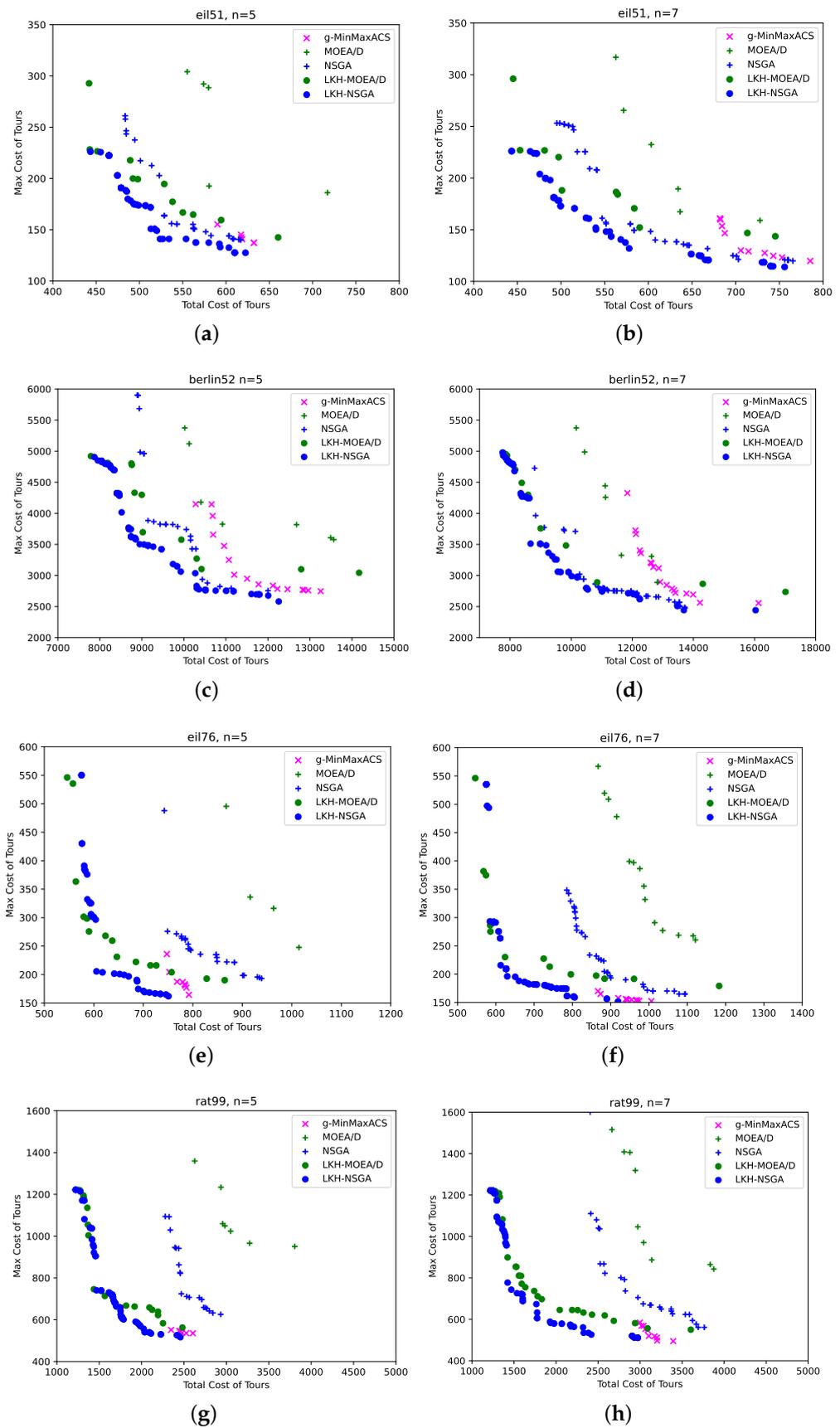


Figure 4. The resulting Pareto fronts of 5 different algorithms on 8 different instances.

Now, we take a closer look at the solution set derived by LKH-guided NSGA. Take the instance *eil51*, $n = 5$ as an example, see LKH-NSGA in Figure 4a. Here, the solution set is formed by 28 individuals, and each is a near-optimal solution. The two ends, i.e., $obj(443.44, 226.08)$ and $obj(622.43, 127.45)$, are the objective values of *minsum* and *minmax* solution, respectively. The *minsum* solution has a total length of 443.44, which is shorter than the *minmax* solution’s 622.43, while the *minmax* solution has a max length of 127.45, which is shorter than the *minsum* solution’s 226.08. Meanwhile, we can observe that for the *minmax* solution, the max length is quite equal to $1/n$ of the total length ($127.45 \approx 622.43/5$), which definitely implies the workloads are well balanced. The other 26 solutions can be viewed as trade-offs of the *minsum* and *minmax* solutions; they together provide sufficient near optimal candidates that can really support decision making.

Table 1. The performance of different algorithms.

| Instances | Indictors | g-MinMaxACS | MOEA/D | NSGA | LKH-MOEA/D | LKH-NSGA | Rel. Impr. LKH-NSGA |
|------------------------------|-----------|-------------|----------|---------------|------------|---------------|---------------------|
| <i>eil51</i> , $n = 5$ | C-metric | 1 | 1 | 1 | 0.786 | 0 | 100% |
| | HV | 0.318 | 0.278 | 0.373 | 0.385 | 0.411 | 6.8% |
| | Spacing | 14.542 | 59.640 | 14.504 | 22.945 | 3.792 | 73.9% |
| <i>eil51</i> , $n = 7$ | C-metric | 1 | 1 | 1 | 0.938 | 0.113 | 88.0% |
| | HV | 0.274 | 0.296 | 0.381 | 0.381 | 0.420 | 10.2% |
| | Spacing | 9.389 | 28.385 | 5.047 | 49.992 | 2.784 | 44.8% |
| <i>berlin52</i> , $n = 5$ | C-metric | 1 | 1 | 1 | 0.812 | 0.011 | 98.6% |
| | HV | 0.272 | 0.214 | 0.306 | 0.303 | 0.342 | 11.8% |
| | Spacing | 118.189 | 1500.125 | 467.152 | 457.896 | 69.930 | 40.8% |
| <i>berlin52</i> , $n = 7$ | C-metric | 0.962 | 1 | 0.670 | 0.862 | 0.220 | 67.2% |
| | HV | 0.328 | 0.332 | 0.407 | 0.401 | 0.445 | 9.3% |
| | Spacing | 189.513 | 1105.016 | 72.243 | 378.840 | 266.841 | - |
| <i>eil76</i> , $n = 5$ | C-metric | 1 | 1 | 1 | 0.600 | 0.595 | 0.8% |
| | HV | 0.359 | 0.217 | 0.325 | 0.399 | 0.431 | 8.0% |
| | Spacing | 13.082 | 107.803 | 42.998 | 17.816 | 5.109 | 60.9% |
| <i>eil76</i> , $n = 7$ | C-metric | 1 | 1 | 1 | 0.615 | 0.232 | 62.3% |
| | HV | 0.315 | 0.196 | 0.327 | 0.402 | 0.437 | 8.7% |
| | Spacing | 9.431 | 71.677 | 8.670 | 70.807 | 6.016 | 30.6% |
| <i>rat99</i> , $n = 5$ | C-metric | 1 | 1 | 1 | 0.647 | 0.222 | 65.7% |
| | HV | 0.399 | 0.223 | 0.367 | 0.503 | 0.526 | 4.6% |
| | Spacing | 35.503 | 200.523 | 209.447 | 66.495 | 15.510 | 56.3% |
| <i>rat99</i> , $n = 7$ | C-metric | 0.8 | 1 | 1 | 0.923 | 0.141 | 82.4% |
| | HV | 0.326 | 0.292 | 0.359 | 0.501 | 0.527 | 5.2% |
| | Spacing | 52.892 | 220.603 | 26.184 | 101.713 | 15.960 | 39.0% |

However, the introduction of an LKH guidance mechanism will inevitably consume more computation time than the raw algorithms. For example, on a desktop computer of Intel Core i9 3.6GHz CPU and 16G RAM, it costs 17.5 s for the raw NSGA and 20.0 s for the LKH-guided NSGA, respectively, on the same instance of *rat99*, $n = 7$. The time consumption of the guidance mechanism highly depends on the parameter of guidance rate p_{LKH} . By increasing the guidance rate, the solution set quality might be improved in the same iteration steps. However, it is not always true because, at the end of the iteration, the guidance mechanism benefits very little to the solution set quality improvement. This phenomenon can be seen in Figure 5, where raw NSGA is shown for clearness and 'LKH-NSGA, $p_{LKH} = 0$ ' means NSGA is initialized by LKH optimized prophet populations but without any guidance during the consequent evolutionary processes. As can be observed, the four different guidance rates, namely $p_{LKH} = 0.001, 0.002, 0.005, 0.01$, make little differ-

ence with respect to the solution set quality. According to our experience, $p_{LKH} = 0.002$ can be a good trade-off between the solution set quality and time consumption.

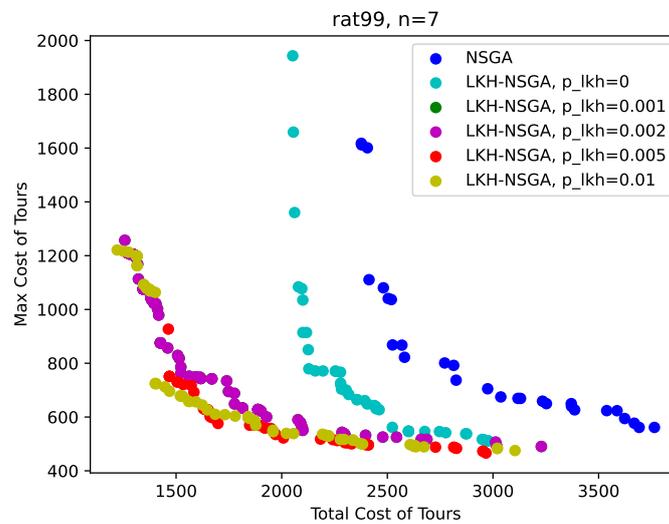


Figure 5. The Pareto fronts of LKH-guided NSGA with different guidance rate p_{LKH} .

6. A General Case Study

A multi-depot MRTA problem (MD-MRTA), where each robot of the swarm departs from its own depot while not from one shared same depot as in the instances of Section 5, is the more general case of MRTA. In this section, we will conduct simulations to show the generalization capability of our proposed LKH-guided NSGA with multi-objective as well as some constraints taken into consideration.

The simulation case is inspired by urban surveying tasks using multiple drones. It is set as follows: $n = 4$ drones are assigned to survey $m = 50$ tasks points that are distributed randomly in a square area of $1000\text{ m} \times 1000\text{ m}$. The objective is to complete the overall tasks as soon as possible while consuming as little energy as possible, given the following two constraints: (a) each drone has a mileage capability of $L_k = 4200\text{ m}$, $\forall k \in \{1, \dots, n\}$, and (b) three of the tasks, for instance schools, are to be visited within specified time-windows. Refer to Figure 6b for a preview of the case, where the drones in their depots are small green-squared and the tasks are red-dotted, three of the tasks, namely T_1, T_5, T_{13} , are with time-windows of $TW_1 = [100\text{ s}, 160\text{ s}]$, $TW_5 = [200\text{ s}, 260\text{ s}]$, $TW_{13} = [210\text{ s}, 270\text{ s}]$, respectively. For simplification, we set the drones traveling in the same velocity $v_k = 5\text{ m/s}$, $\forall k \in \{1, \dots, n\}$; thus, the bi-objective can be simplified as a minimization of the maximum cost of the tours and the total cost of the tours, simultaneously.

As proposed, the depots of R_1, \dots, R_4 are treated as virtual tasks that can form the one-chromosome coding of the solution. Then, the LKH-guided NSGA is performed to solve the case, resulting in the Pareto front, as shown in Figure 6a. Every dot in the Pareto front represents a near-optimal solution that can serve as a candidate for decision making. To visualize, we sample three from the solution set, see sub Figure 6b–d that correspond to objective points (b)–(d) in Figure 6a, where Figure 6b is the *minsum* solution and Figure 6c is the *minmax* solution. The *minsum* solution achieves a total cost of tours of 7873.50 m, which is the optimal until now, but the lack of cooperation among robots (some robots take most of the tasks while some take very little) will inevitably lead to a long maximum cost of tours and thus the long overall completion time. The case is vice versa for the *minmax* solution, where the workload among robots is relatively balanced, resulting in the shortest maximum sub-tour of 2493.01 m, so that the overall completion time is optimized, but the total cost of tours of 8937.50 m is the highest among the solution set. As a trade-off, the decision maker can also choose another solution like (d) in Figure 6a depending on the decision preference. The task allocation and each robot’s route planning of this solution are shown in Figure 6d,

the maximum cost of sub-tours increases by only 21.14 m (=2514.15 – 2493.01), but the total cost will save 414.80 m (=8937.50 – 8522.70) compared to the solution of Figure 6c.

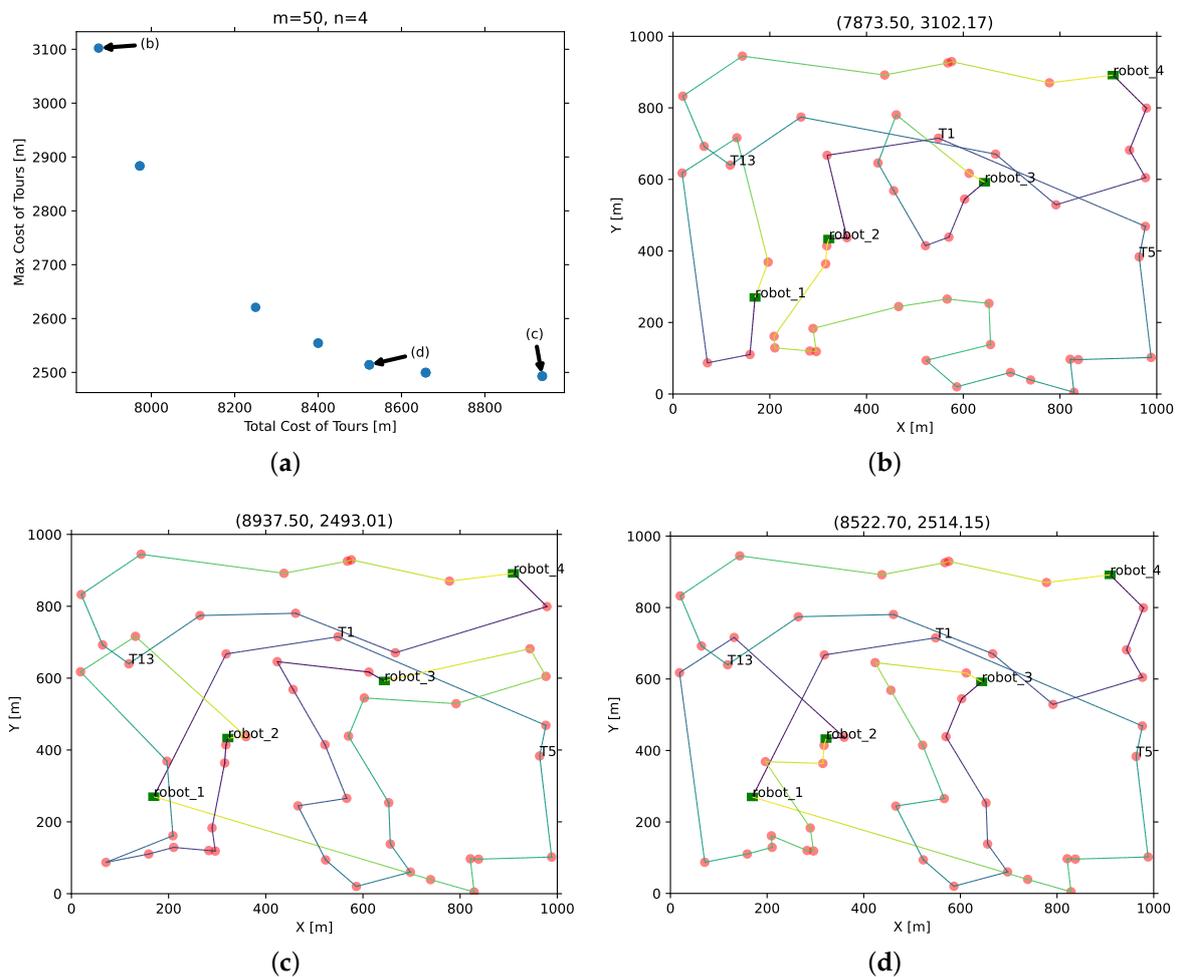


Figure 6. An instance of MD-MRTA with constraints: (a) Pareto front, (b–d) different solutions of task allocation and route planning for each robot.

Figures 7 and 8 show the constraints fulfillment of the solution set; the three sub-figures correspond to the three sampled solutions from Figure 6a. It is clear in Figure 7 that all the robots’ travel distances are below the mileage capability of each robot. Figure 8 illustrates the time sequences of the assigned tasks of each robot, where the tasks id. are shown alongside the timeline, three tasks with time-window constraints, namely T_1 , T_5 , T_{13} , are highlighted above the task, each with a couple of small triangles to indicate the required start time and end time. It can be observed that the moment the robot reached the tasks is right within the predefined time-windows.

In summary, our approach can provide a satisfactory (near optimal) solution set that can really help practical decision making for multi-objective MRTA problems. Once the final solution is confirmed by the decision maker, the corresponding task planning can be downloaded into each robot. A robot would then execute its allocated tasks in sequence, as illustrated by the color-mapped lines in Figure 6b–d; i.e., it would move from the dark black end to the bright yellow end.

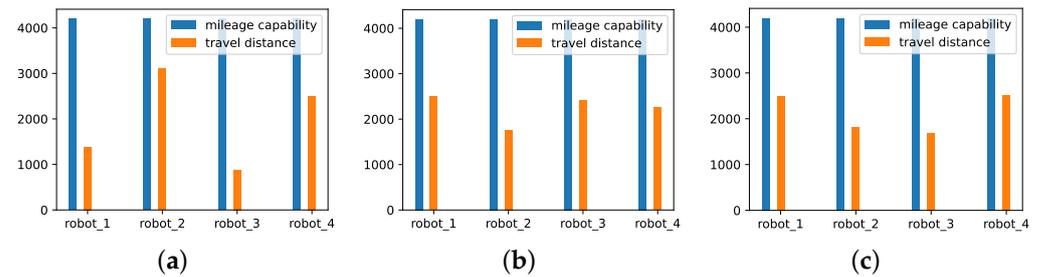


Figure 7. The travel distance of each robot, (a–c) correspond to the three solutions sampled from Figure 6a.

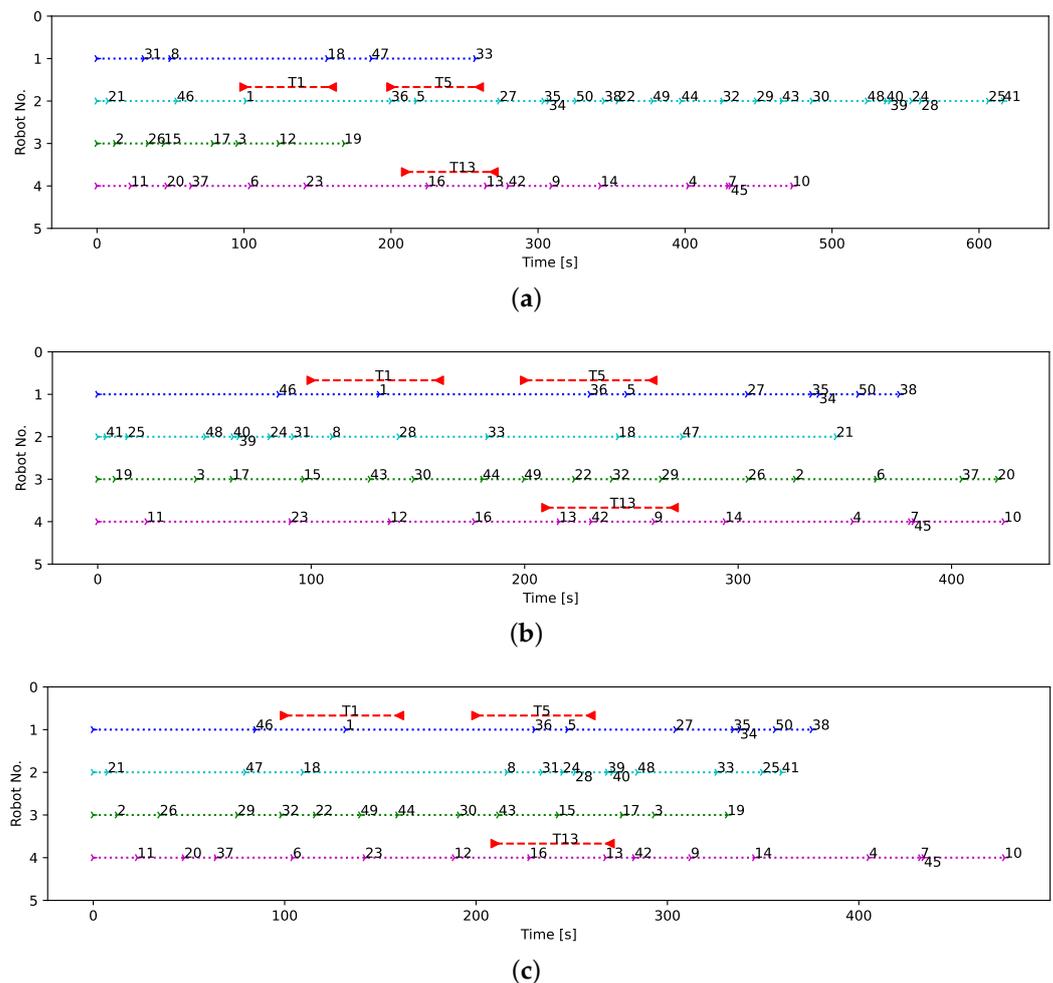


Figure 8. The time sequence the assigned tasks of each robot, (a–c) correspond to the three solutions sampled from Figure 6a.

7. Conclusions

The paper aims to resolve the Pareto set of multi-objective MRTA and route-planning problem in an efficient way so as to assist in practical decision making. The proposed one-chromosome representation technique to code a solution is concise and can ease the succeeding genetic operations. The cost matrix is extended accordingly to serve as a look-up table to the following processes of multi-objective evolutionary algorithms. The LKH solver is employed as a highly efficient sub-tour planner to generate prophet generation at the beginning of the evolution as well as to guide the evolutionary direction during the proceeding of the evolutionary algorithms. The LKH guidance mechanism has been

proven to be effective in the conducted numerical experiments, both for MOEA/D and NSGA, where the latter holds better performance on almost all the indicators, i.e., *C-metric*, *HV*, and *Spacing*. The guidance rate of the LKH guidance plays an important role in the performance of the algorithm by increasing the rate the solution quality will be improved to some extent; however, the benefit diminishes when the rate is higher than a certain value. At last, with the benefit of the one-chromosome representation technique and the LKH-guided NSGA algorithm, the more general instance of multi-objective MD-MRTA with constraints can be resolved efficiently, and the resulting Pareto set can serve as a powerful tool to practical decision making.

The limitation of the work lies in the homogeneous diagram of the researched multi-robot systems; the heterogeneous diagram is also an interesting topic and will be studied in our future work. Another challenging work is dynamic task allocation after the robots are sent to execute the tasks in a dynamic environment, which relates to dynamic programming, distributed cooperation, etc.

Author Contributions: Conceptualization, X.J. and Z.Z.; methodology, Z.Z.; software, Z.Z.; validation, Z.Z.; formal analysis, X.J.; investigation, Z.Z.; resources, S.M.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z.; visualization, X.J.; supervision, S.M.; project administration, Z.Z.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Key Research and Development Program of Shandong Province (2021CXGC011304); Project of Natural Science Foundation of Shandong Province (ZR2020MF066); Youth Innovation Group Project of Shandong University (2020QNQT016); China University Innovation Fund (2021ZYA12004).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feroz, A.K.; Zo, H.; Chiravuri, A. Digital transformation and environmental sustainability: A review and research agenda. *Sustainability* **2021**, *13*, 1530. [\[CrossRef\]](#)
2. Hanelt, A.; Bohnsack, R.; Marz, D.; Antunes Marante, C. A systematic review of the literature on digital transformation: Insights and implications for strategy and organizational change. *J. Manag. Stud.* **2021**, *58*, 1159–1197. [\[CrossRef\]](#)
3. Delaram, J.; Houshamand, M.; Ashtiani, F.; Valilai, O.F. A utility-based matching mechanism for stable and optimal resource allocation in cloud manufacturing platforms using deferred acceptance algorithm. *J. Manuf. Syst.* **2021**, *60*, 569–584. [\[CrossRef\]](#)
4. Delaram, J.; Houshamand, M.; Ashtiani, F.; Fatahi Valilai, O. Development of public cloud manufacturing markets: A mechanism design approach. *Int. J. Syst. Sci. Oper. Logist.* **2022**, 1–27. [\[CrossRef\]](#)
5. Dorigo, M.; Theraulaz, G.; Trianni, V. Reflections on the future of swarm robotics. *Sci. Robot.* **2020**, *5*, eabe4385. [\[CrossRef\]](#)
6. Verma, J.K.; Ranga, V. Multi-robot coordination analysis, taxonomy, challenges and future scope. *J. Intell. Robot. Syst.* **2021**, *102*, 10. [\[CrossRef\]](#)
7. Cheng, Y.; Sun, F.; Zhang, Y.; Tao, F. Task allocation in manufacturing: A review. *J. Ind. Inf. Integr.* **2019**, *15*, 207–218. [\[CrossRef\]](#)
8. Brambilla, M.; Ferrante, E.; Birattari, M.; Dorigo, M. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **2013**, *7*, 1–41. [\[CrossRef\]](#)
9. Farinelli, A.; Boscolo, N.; Zanutto, E.; Pagello, E. Advanced approaches for multi-robot coordination in logistic scenarios. *Robot. Auton. Syst.* **2017**, *90*, 34–44. [\[CrossRef\]](#)
10. Zheng, H.; Yuan, J. An Integrated Mission Planning Framework for Sensor Allocation and Path Planning of Heterogeneous Multi-UAV Systems. *Sensors* **2021**, *21*, 3557. [\[CrossRef\]](#)
11. Zaidi, L.; Sahnoun, M.; Bettayeb, B. Task allocation based on shared resource constraint for multi-robot systems in manufacturing industry. *IFAC-PapersOnLine* **2019**, *52*, 2020–2025. [\[CrossRef\]](#)
12. Wei, Z.; Xia, C.; Yuan, X.; Sun, R.; Lyu, Z.; Shi, L.; Ji, J. The path planning scheme for joint charging and data collection in WRSNs: A multi-objective optimization method. *J. Netw. Comput. Appl.* **2020**, *156*, 102565. [\[CrossRef\]](#)
13. Khamis, A.; Hussein, A.; Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 31–51.
14. Gerkey, B.P.; Mataric, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [\[CrossRef\]](#)
15. Koubâa, A.; Bennaceur, H.; Chaari, I.; Trigui, S.; Ammar, A.; Sriti, M.F.; Alajlan, M.; Cheikhrouhou, O.; Javed, Y. *Robot Path Planning and Cooperation: Foundations, Algorithms and Experimentations*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 772.

16. Tkach, I.; Edan, Y. *Distributed Heterogeneous Multi Sensor Task Allocation Systems*; Springer: Berlin/Heidelberg, Germany, 2020.
17. Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* **2006**, *94*, 1257–1270. [[CrossRef](#)]
18. Kivelevitch, E.; Cohen, K.; Kumar, M. A market-based solution to the multiple traveling salesmen problem. *J. Intell. Robot. Syst.* **2013**, *72*, 21–40. [[CrossRef](#)]
19. Wei, C.; Ji, Z.; Cai, B. Particle swarm optimization for cooperative multi-robot task allocation: A multi-objective approach. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2530–2537. [[CrossRef](#)]
20. Jose, K.; Pratihari, D.K. Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robot. Auton. Syst.* **2016**, *80*, 34–42. [[CrossRef](#)]
21. Liu, H.; Li, X.; Wu, G.; Fan, M.; Wang, R.; Gao, L.; Pedrycz, W. An iterative two-phase optimization method based on divide and conquer framework for integrated scheduling of multiple UAVs. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5926–5938. [[CrossRef](#)]
22. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [[CrossRef](#)]
23. Cheikhrouhou, O.; Koufi, I. A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy. *Comput. Sci. Rev.* **2021**, *40*, 100369. [[CrossRef](#)]
24. Carter, A.E.; Ragsdale, C.T. A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *Eur. J. Oper. Res.* **2006**, *175*, 246–257. [[CrossRef](#)]
25. Yuan, S.; Skinner, B.; Huang, S.; Liu, D. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. *Eur. J. Oper. Res.* **2013**, *228*, 72–82. [[CrossRef](#)]
26. Király, A.; Abonyi, J. Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm. In *Intelligent Computational Optimization in Engineering*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 241–269.
27. Lupoai, V.I.; Chili, I.A.; Breaban, M.E.; Raschip, M. SOM-guided evolutionary search for solving minmax multiple-TSP. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 73–80.
28. Zhou, B.; Zhou, R.; Gan, Y.; Fang, F.; Mao, Y. Multi-Robot Multi-Station Cooperative Spot Welding Task Allocation Based on Stepwise Optimization: An Industrial Case Study. *Robot.-Comput.-Integr. Manuf.* **2022**, *73*, 102197. [[CrossRef](#)]
29. Mahmud, M.S.A.; Abidin, M.S.Z.; Buyamin, S.; Emmanuel, A.A.; Hasan, H.S. Multi-objective route planning for underwater cleaning robot in water reservoir tank. *J. Intell. Robot. Syst.* **2021**, *101*, 9. [[CrossRef](#)]
30. Coello, C.C. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–36. [[CrossRef](#)]
31. Przybylski, A.; Gandibleux, X. Multi-objective branch and bound. *Eur. J. Oper. Res.* **2017**, *260*, 856–872. [[CrossRef](#)]
32. Benson, H. Further analysis of an outcome set-based algorithm for multiple-objective linear programming. *J. Optim. Theory Appl.* **1998**, *97*, 1–10. [[CrossRef](#)]
33. Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; TIK-Report 103; ETH Zurich: Zurich, Switzerland, 2001.
34. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
35. Jain, H.; Deb, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **2013**, *18*, 602–622. [[CrossRef](#)]
36. Beume, N.; Naujoks, B.; Emmerich, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **2007**, *181*, 1653–1669. [[CrossRef](#)]
37. Erfani, T.; Utyuzhnikov, S.V.; Kolo, B. A modified directed search domain algorithm for multiobjective engineering and design optimization. *Struct. Multidiscip. Optim.* **2013**, *48*, 1129–1141. [[CrossRef](#)]
38. Bektas, T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* **2006**, *34*, 209–219. [[CrossRef](#)]
39. Necula, R.; Breaban, M.; Raschip, M. Tackling the bi-criteria facet of multiple traveling salesman problem with ant colony systems. In Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, 9–11 November 2015; pp. 873–880.
40. Lu, L.C.; Yue, T.W. Mission-oriented ant-team ACO for min-max MTSP. *Appl. Soft Comput.* **2019**, *76*, 436–444. [[CrossRef](#)]
41. Al-Omeer, M.A.; Ahmed, Z.H. Comparative study of crossover operators for the MTSP. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 3–4 April 2019; pp. 1–6.
42. Helsgaun, K. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Eur. J. Oper. Res.* **2000**, *126*, 106–130. [[CrossRef](#)]
43. Li, M.; Yao, X. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Comput. Surv. CSUR* **2019**, *52*, 26. [[CrossRef](#)]