EFFICIENT DATA ASSOCIATION ALGORTIHMS

EFFICIENT DATA ASSOCIATION ALGORITHMS FOR MULTITARGET TRACKING

BY

JINGQUN LI, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2019) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Efficient data association algorithms for multitarget tracking
AUTHOR:	Jingqun Li M.Sc., (Applied Mathematics and Physics) Beijing University of Aeronautics and Astronautics, Beijing, China
SUPERVISOR:	Thia Kirubarajan
NUMBER OF PAGES:	xvi, 151

Abstract

In this work, we propose a novel convex dual approach to the multidimensional dimensional assignment problem, which is an NP-hard binary programming problem. It is shown that the proposed dual approach is equivalent to the Lagrangian relaxation method in terms of the best value attainable by the two approaches. However, the pure dual representation is not only more elegant, but also makes the theoretical analysis of the algorithm more tractable. In fact, we obtain a sufficient and necessary condition for the duality gap to be zero, or equivalently, for the Lagrangian relaxation approach to find the optimal solution to the assignment problem with a guarantee. Also, we establish a mild and easy-to-check condition, under which the dual problem is equivalent to the original one. In general cases, the optimal value of the dual problem can provide a satisfactory lower bound on the optimal value of the original assignment problem.

We then extend the purely dual formulation to handle the more general multidimensional assignment problem. The convex dual representation is derived and its relationship to the Lagrangian relaxation method is investigated once again. Also, we discuss the condition under which the duality gap is zero. It is also pointed out that the process of Lagrangian relaxation is essentially equivalent to one of relaxing the binary constraint condition, thus necessitating the auction search operation to

recover the binary constraint. Furthermore, a numerical algorithm based on the dual formulation along with a local search strategy is presented.

Finally, the newly proposed algorithm is shown to outperform the Lagrangian relaxation method in a number of multitarget tracking simulations.

To the memory of Mr. Chengxiao Jin,

my former math teacher

Acknowledgements

My thanks first go to Professor Thia Kirubarajan for providing me with the opportunity of studying multi-target tracking and conducting research in this exciting field. I am grateful to his guide and help in the three years of study at McMaster.

I have also benefited from the help and support of Professor Krishna R. Pattipati and Dr R. Tharmarasa to my research. Also, I want to thank Professor Aleksandar Jeremic and Professor Tim Field for their valuable advice to my study.

Thank the colleagues in the estimation, tracking and fusion laboratory for their help. In particular, thank Yinghui and Ben for discussions on our research field.

Moreover, I am grateful to Ms Cheryl Gies for her help during the three years.

Last but not least I want to thank my dear parents, sister and brother, my wife and my daughter for their love and support to my research.

Contents

List of Figures

List of Tables

List of symbols for Chapter 2

N	The size of each dimension
I_0	The index set $\{1, ..., N\} \times \{1, ..., N\} \times \{1, ..., N\}$
$c = (c_{ijk})$	Costs array, with $(i, j, k) \in I_0$
$f(x)$	The objective function of 3-d assignment problems, and is equal to $\sum_{(i,j,k)\in I_0} x_{ijk} c_{ijk}$
$\mu = (\mu_1, \ldots, \mu_N)$	Lagrangian multipliers
$\nu = (\nu_1, \ldots, \nu_N)$	Lagrangian multipliers
$\eta = (\eta_1, \ldots, \eta_N)$	Lagrangian multipliers
$H(\mu, \nu, \eta)$	Dual objective function
$G(\eta)$	The objective function of the Lagrangian relaxation method
J^*_{LR}	The optimal value of $G(\eta)$

List of symbols for Chapter 3

M	The number of dimensions of the assignment problem
N	The size of each dimension
I_0	The index set $\{1, ..., N\}^M$
$c = (c_{i_1 \ldots 1_M})$	Costs array, with $(i_1, \ldots, i_M) \in I_0$
$f(x)$	The objective function of M-dimensional assignment problems, and is equal to $\sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1 \ldots 1_M} x_{i_1 \ldots i_M}$
μ	Lagrangian multiplier, abbreviation for $(\mu_1, \ldots, \mu_N)$, with each $\mu_i = (\mu_{i,1}, \mu_{i,2}, ..., \mu_{i,N})$
μ^*	A particular value of μ, usually representing an optimal solution to the dual problem
$H(\mu)$	Dual objective function
H^*	The optimal value of $H(\mu)$
$G^{[1]}, \ldots, G^{[M-1]}$	The objective functions of the Lagrangian relaxation method
J^*_{LR}	The optimal value of $G^{[M-1]}$

List of symbols for Chapter 4

P_F	False alarm probability
P_D	The probability of detection
$z_j(k)$	Measurement j at time k
$\Lambda_{i,j}(k)$	The probability of associating the measurement $z_j(k)$ to target i at time k
$\Lambda_{i,j,m}(k)$	The probability of associating the measurements $z_j(k)$ and $z_m(k+1)$ to target i at time k
$c_{i_1 \ldots 1_M}$	The assignment costs, and are the negative log-likelihood ratios of the $M-1$ frame association problem
$f(\rho)$	The objective function of M-dimensional assignment problems, and is equal to $\sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1 \ldots 1_M} \rho_{i_1 \ldots i_M}$
μ	Lagrangian multiplier, abbreviation for $(\mu_1, \ldots, \mu_N)$, with each $\mu_i = (\mu_{i,1}, \mu_{i,2}, ..., \mu_{i,N})$
$L(\mu)$	Dual objective function
F	State equation coefficients matrix

H	Linearized observation equation coefficients matrix

Acronyms

EKF	Extended Kalman filter
JPDA	Joint Probabilistic Data Association
LP	Linear programming
LR	Lagrangian relaxation
MAP	Multidimensional assignment problem
MHT	Multiple hypothesis tracking
PDA	Probabilistic Data Association

Chapter 1

Background and Motivation

1.1 Data association problem

Data association is an important step in tracking algorithms. The task of data association is to assign the measurements to the tracks in the system. That is, data association determines the origin of each measurement. In the simplest case of single target tracking, only a single target is to be tracked, and the nearest neighborhood data association method is the most commonly used algorithm. However, for multi-target tracking, in which multiple targets are to be tracked with observations possibly from multiple platforms, we need to develop efficient data association algorithms. Here, the difficulty is to handle the exponential increase in the computational load with the number of targets and measurements.

In this introduction chapter, we first review the nearest neighborhood association algorithm. The commonly used Joint Probabilistic Data Association (JPDA) algorithm is then described in full detail. Next, we study the data association algorithm based on S-D assignment, in which $S-1$ frames of measurements are to be assigned

to the targets being tracked, and point out the difficulties in these methods and our approach to address these difficulties.

1.2 Nearest neighbor association [4]

For tracking a single target, the nearest neighborhood association algorithm is frequently applied. In this algorithm, a region commonly referred to as a gate is identified such that only the measurements inside the region are considered for data association. The region is given by

$$\{z|D(z) < r\}, \tag{1.2.1}$$

where r is an adjustable parameter taking different values corresponding to different confidence levels, z is the sensor measurement, and $D(z)$ is the square of a normalized distance between z and the predicted measurement $\hat{z}$:

$$D(z) = [z - \hat{z}(k|k-1)]' S(k)^{-1} [z - \hat{z}(k|k-1)], \tag{1.2.2}$$

where $S(k)$ is the covariance of the innovation, which is the difference between the actual measurement and the predicted one.

The measurement in the gate with the smallest value $D(z)$ is chosen to be the measurement associated to the target.

Another commonly used data association algorithm for tracking a single target is Probabilistic Data Association (PDA) [4], which has been extended into the JPDA algorithm for tracking multiple targets.

1.3 Joint Probabilistic Data Association (JPDA) algorithm [4]

If the number of targets is known and relatively small, and the clutter, which is the noise caused by the environment, is not heavy, then the JPDA algorithm can be applied to perform data association, as well as the estimation, also referred to as filtering in the literature.

1.3.1 Joint association event probabilities

First, we need to calculate the probability of a joint association event

$$\theta(k) = \cap_{j=1}^{m(k)} \theta_{jt_j}(k), \tag{1.3.1}$$

where

$$m(k) = \#Z(k), \tag{1.3.2}$$

the number of measurements in $Z(k)$, the kth data set, and $\theta_{jt_j}(k)$ is the event that the jth measurement in $Z(k)$ is associated with the t_jth target.

Let the accumulative measurements up to time k be denoted by $Z^k = \{Z(1), \ldots, Z(k)\}$. The probability of the joint association event, given the accumulative measurements, is

$$\begin{aligned} \mathbb{P}(\theta(k)|Z^k) &= \mathbb{P}\big(\theta(k)|Z(k), m(k), Z^{k-1}\big) \\ &= \frac{1}{c}\mathbb{P}\big(\theta(k), Z(k)|m(k), Z^{k-1}\big), \end{aligned} \tag{1.3.3}$$

where

$$c = \mathbb{P}\big(Z(k)|m(k), Z^{k-1}\big). \tag{1.3.4}$$

Furthermore,

$$\begin{aligned} &\mathbb{P}\big(\theta(k), Z(k)|m(k), Z^{k-1}\big) \\ &= p\big(Z(k)|\theta(k), m(k), Z^{k-1}\big)\mathbb{P}(\theta(k)|m(k), Z^{k-1}). \end{aligned} \tag{1.3.5}$$

Under the independence assumption, the first term in Eq.(1.3.5) is

$$\begin{aligned} &p\big(Z(k)|\theta(k), m(k), Z^{k-1}\big) \\ &= \prod_{j=1}^{m(k)} p\big(z_j(k)|\theta_{jt_j}(k), Z^{k-1}\big) \\ &= V^{-\phi} \prod_{j=1}^{m(k)} f_{t_j}\big(z_j(k)\big)^{\tau_j}, \end{aligned} \tag{1.3.6}$$

where

$$\tau_j = \begin{cases} 1, \text{ if measurement } j \text{ is associated to target } t_j, \\ 0, \text{otherwise}, \end{cases} \tag{1.3.7}$$

since

$$p\big(z_j(k)|\theta_{jt_j}(k), Z^{k-1}\big) = \begin{cases} f_{t_j}\big(z_j(k)\big), \text{ if measurement } j \text{ is associated to target } t_j, \\ V^{-1}, \text{otherwise}, \end{cases} \tag{1.3.8}$$

where V is the volume of the observation field, and

$$f_{t_j}\big(z_j(k)\big) = \mathcal{N}[z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k)], \tag{1.3.9}$$

the normal distribution density with the mean equal to $\hat{z}_{t_j}(k|k-1)$, the predicted measurement for target t_j, and the covariance equal to $S_{t_j}(k)$], the innovation covariance.

On the other hand, by a combinatorial argument, the second term in Eq.(1.3.5) is found to be

$$\begin{aligned}
&\mathbb{P}(\theta(k)|m(k), Z^{k-1}) \\
&= \frac{\phi!}{m(k)!}\mathbb{P}(\text{there are } \phi \text{ false alarms in the } m(k) \text{ measurements}|m(k), Z^{k-1}) \\
&= \frac{\phi!}{m(k)!}\Big[\prod_t (P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}\Big]\frac{\mu_F(\phi)}{\mathbb{P}(m(k))},
\end{aligned} \tag{1.3.10}$$

where

$$\delta_t = \begin{cases} 1, \text{ if a measurement } j \text{ is associated to target } t_j \text{ by } \theta(k), \\ 0, \text{otherwise}, \end{cases} \tag{1.3.11}$$

P_D^t is the detection probability of target t, ϕ is the number of measurements associated to false alarms in $\theta(k)$, and $\mu_F(\phi)$ is the prior probability mass function (pmf), that is, the known probability distribution of false alarms. Such a distribution depends on the nature of the clutter and is described by the clutter model. For more information on clutter models, see [4, 43]).

For the parametric JPDA, we use Poisson pmf for the clutter model:

$$\mu_F(\phi) = e^{-\lambda V}\frac{(\lambda V)^\phi}{\phi!}. \tag{1.3.12}$$

Combining this model with Eq.(1.3.10), and Eq.(1.3.5), we get

$$\begin{aligned}&\mathbb{P}(\theta(k)|Z^k)\\&=\frac{1}{c_1}\prod_{j=1}^{m(k)}\left\{\lambda^{-1}f_{t_j}\left[z_j(k)\right]\right\}^{\tau_j}\left[\prod_t (P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}\right],\end{aligned}\tag{1.3.13}$$

where c_1 is a normalizing constant making the total probability equal to 1.

There is also a nonparametric clutter model, in which

$$\mu_F(\phi)=\epsilon,\tag{1.3.14}$$

a constant determined by the volume of the observation field.

With this model, we similarly get

$$\begin{aligned}&\mathbb{P}(\theta(k)|Z^k)\\&=\frac{\phi!}{c_2}\prod_{j=1}^{m(k)}\left\{Vf_{t_j}\left[z_j(k)\right]\right\}^{\tau_j}\left[\prod_t (P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}\right].\end{aligned}\tag{1.3.15}$$

1.3.2 The state estimation

Now, the marginal association probabilities

$$\beta_{jt}=\mathbb{P}\big(\theta_{jt}|Z^k\big)\tag{1.3.16}$$

can be derived from the joint probabilities:

$$\beta_{jt}=\sum_{\theta:\theta_{jt}\in\theta}\mathbb{P}\big(\theta_{jt}|Z^k\big).\tag{1.3.17}$$

In order to estimate the states of the targets, we need one filter for each of the targets (the number of the targets is known).

The state estimation equation of the tth target is given by

$$\hat{x}^t(k|k) = \hat{x}^t(k|k-1) + W^t(k)\nu^t(k), \tag{1.3.18}$$

where we use the superscript t to represent the filter for the tth target, and

$$\nu^t(k) = \sum_{j=1}^{m(k)} \beta_{jt}(k)\nu_{jt}(k), \tag{1.3.19}$$

with

$$\nu_{jt}(k) = z_j(k) - \hat{z}^t(k|k-1), \tag{1.3.20}$$

and

$$W^t(k) = P^t(k|k-1)\big(H^t(k)\big)'\big(S^t(k)\big)^{-1}. \tag{1.3.21}$$

The covariance associated with the updated state of the tth target is

$$P^t(k|k) = \beta_{0t}(k)P^t(k|k-1) + [1-\beta_{0t}(k)]P^{tc}(k|k) + \tilde{P}^t(k) \tag{1.3.22}$$

where the covariance of the state updated with the corrected measurement is

$$P^{tc}(k|k) = P^t(k|k-1) - W^t(k)S^t(k)\big(W^t(k)\big)' \tag{1.3.23}$$

and the spread of the innovation term is

$$\tilde{P}^t(k) = W^t(k)\Big[\sum_{j=1}^{m(k)} \beta_{jt}(k)\nu_{jt}(k)\nu_{jt}(k)' - \nu^t(k)(\nu^t(k))'\Big]\big(W^t(k)\big)' \tag{1.3.24}$$

1.4 S-D assignment based data association [4, 11]

Suppose we have S sensors. The ith measurement from sensor s is denoted by $z_i(s)$. Then a set of measurements will be received at each scan instant. We are faced with the problem of associating the measurements to targets being tracked. A class of association algorithm can be developed based on combinatorial optimization, in which the association problem is formulated as a linear assignment problem.

Similar to Eq.(1.3.13) or Eq.(1.3.15), we can define the likelihood function that the measurement S-tuple $Z_{i_1 i_2 \ldots 1_S} = \big(\mathbf{z}_{i_1}(1), \mathbf{z}_{i_1}(2), \ldots, \mathbf{z}_{i_S}(S)\big)$ is from a target t:

$$\begin{aligned} &\Lambda(Z_{i_1 i_2 \ldots 1_S}|t) \\ &= \prod_{s=1}^{S} \big\{(P_D^s p\big[\mathbf{z}_{i_s}(s)|\mathbf{x}_t\big]\big\}^{u(i_s)}(1-P_D^s)^{1-u(i_s)}, \end{aligned} \tag{1.4.1}$$

where

$$u(i_s) = \begin{cases} 0, \text{ if } i_s = 0 \text{ (the target is missing)}, \\ 1, \text{otherwise}, \end{cases} \tag{1.4.2}$$

Since $\mathbf{x}_t$ is unknown in Eq.(1.4.1), and hence will be replaced by its ML estimate:

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} \Lambda(Z_{i_1 i_2 \ldots 1_S}|t) \tag{1.4.3}$$

The likelihood that the measurements are all spurious (i.e. $t = \emptyset$) is

$$\Lambda(Z_{i_1 i_2 \ldots 1_S} | t = \emptyset) = \prod_{s=1}^{S} \big[\frac{1}{\Psi_s}\big]^{u(i_s)}, \tag{1.4.4}$$

where Ψ_s is the false alarm probability in a cell multiplied by the volume of the resolution cell.

A association cost of associating $Z_{i_1 i_2 \ldots 1_S}$ to target t is then defined by:

$$\begin{aligned} c_{i_1 i_2 \ldots i_S} = \sum_{s=1}^{S} \Big\{ & [u(i_s) - 1] \ln(1 - P_D^s) \\ & - u(i_s) \ln\big(\frac{P_D^s \Psi_s}{|2\pi\Sigma_s|^{1/2}}\big) + u(i_s) \\ & \Big(\frac{1}{2}\big[\mathbf{z}_{i_s}(s) - H(\hat{\mathbf{x}}_t, \mathbf{y}_{i_s}(s))\big]^T \Sigma_s^{-1} \\ & \big[\mathbf{z}_{i_s}(s) - H(\hat{\mathbf{x}}_t, \mathbf{y}_{i_s}(s))\big]\Big)\Big\} \end{aligned} \tag{1.4.5}$$

With the above costs, the data association problem will be formulated as an optimization problem, more specifically, a multidimensional assignment problem:

$$\min_{\rho} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S}, \tag{1.4.6}$$

subject to the constraints:

$$\rho_{i_1 i_2 \ldots i_S} \in \{0, 1\} \tag{1.4.7}$$

$$\sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \forall i_1 \in 1, ..., n_1, \tag{1.4.8}$$

$$\sum_{i_1=0}^{n_1} \sum_{i_3=0}^{n_3} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \forall i_2 \in 1, ..., n_2, \tag{1.4.9}$$

$$\ldots \tag{1.4.10}$$

$$\sum_{i_1=0}^{n_1} \cdots \sum_{i_{S-1}=0}^{n_{S-1}} \rho_{i_1 i_2 \ldots i_S} = 1, \forall i_S \in 1, ..., n_S. \tag{1.4.11}$$

In the above formulation, measurement 0 in each frame represents a missing target, and

$$\rho_{i_1 i_2 \ldots i_S} = \begin{cases} 1 & \text{if measurements } Z_{i_1 i_2 \ldots {}_1 S} \text{ are associated to a target,} \\ 0 & \text{otherwise.} \end{cases}$$

These constraints specify that each real measurement has to be assigned to only one measurement in any other frame.

1.5 Review of the assignment algorithm

In the previous section, we see that the data association problem arising in the context of multisensor multitarget tracking applications can be formulated as a Multidimensional assignment problem (MAP). The MAP is an extension of the classical 2-D linear assignment problem. See [36] for a detailed account of assignment problems arising in multiple-target tracking applications. Moreover, the MAP approach is an efficient way to implement the Multiple hypothesis tracking (MHT) algorithm [8],

which is a key algorithm in autonomous driving vehicles [1] and surveillance systems [17].

In the classical 2-D assignment problem, m persons are to be matched to m objects (or to $n > m$ objects in asymmetric assignment problems) such that the total benefit is maximized [7]. It can be used to model a wide variety of practical applications, such as resource assignment problems for wireless communication [32]. The research on 2-D assignment problem has a relatively long history, see [34] for an extensive review of different types of assignment problems. As a result, there exist some well known algorithms to find optimal solutions to the 2-D problem efficiently, namely, in polynomial time. The Hungarian method was proposed in [26]. Several versions of the auction algorithm were proposed in [7]. The signature method was developed in [2]. A shortest augmenting path algorithm was presented in [21]. A review of these as well as other methods can be found in [12, 9].

The MAP, however, has been shown to be NP-hard [16]. As a consequence, exact algorithms for MAP, such as the branch and bound algorithm, turn out to be quite inefficient, especially for large-size problems [38]. Approximate MAP algorithms have to be used in some practical applications such as object tracking for surveillance involving real-time decision making.

Approximate methods for the MAP with decomposable costs are considered in [3] and are further investigated in [44] using a graph partition method. Integer programming models for the MAP with star costs are presented in [45]

Heuristic algorithms have been widely studied for MAPs as well, see[22] for a survey and generalizations of heuristic approaches, including construction heuristics, metaheuristics and local search heuristics. Other related methods for MAPs include

the greedy method [39], and a solver for multi-index assignment presented in [35], which integrated the idea of cutting-planes and heuristics into a branch-and-cut algorithm.

Recently, solutions based on deep learning have been proposed for some conventional optimization problems. Machine learning methods to solve assignment problems in multi-target tracking are surveyed in [13], where a review of probabilistic graphical models for multiple dimensional assignment problems, together with the belief propagation method and the Markov chain Monte Carlo method are presented. The deep learning method is discussed as well. Moreover, it is pointed out that reinforcement learning approach is preferred over supervised learning approaches to avoid the requirement of generating MAPs with known solutions, which is hard to get due to the NP-hardness of MAPs.

On the other hand, a Lagrangian relaxation (LR) method was proposed to solve three dimensional assignment problems in [15]. Later, the method was extended to solve MAPs by [33, 37, 11, 38]. Another related work is a dual decompostion method in [27].

1.6 Objective and organization of the thesis

The purpose of this research is to find an efficient algorithm for multidimensional assignment problems, in order to tackle the MAPs arising from the data association operations in multi-target tracking applications.

The thesis is organized as follows. We first propose a novel approach for three dimensional assignment problems in Chapter 2. In order to assess the efficiency of the new method, we generate a series of random assignment problems with known optimal

solutions. The dual method is then applied to solve these problems. Meanwhile, the LR method is applied to these problem as well, to serve as a benchmark for the performance evaluation. These simulations show that the dual approach outperforms the LR method.

The dual assignment approach is extended to solve more general multidimensional problems in Chapter 3. The convex dual representation is first derived and its relationship to the LR method is investigated. Also, we discuss the condition under which the duality gap is zero. We also point out that the process of LR is essentially equivalent to relaxing the binary constraint condition, thus necessitating the auction search operation to recover the binary constraint. Furthermore, a numerical algorithm based on the dual formulation along with a local search strategy is presented. Again, the simulation results show that the proposed algorithm has better performances than the LR method.

In chapter 4, we will first formulate the data association operation in multi-frame multi-target tracking as a multidimensional assignment problem. Then, the newly proposed dual assignment method will be applied to solve the resulting assignment problems. The simulation results show that the association algorithm is effective and efficient in complex tracking scenarios.

1.7 Publications

Li, J., Tharmarasa, R., Brown, D., Kirubarajan, T., and Pattipati, K. R. A novel convex dual approach to three-dimensional assignment problem: Theoretical analysis. Computational Optimization and Applications. On line access: https://doi.org/10.1007/s10589-019-00113-w, Published on May 28, 2019.

Li, J., Kirubarajan, T., Tharmarasa, R., Brown, D., and Pattipati, K. R.: Dual algorithm for multi-dimensional assignment problems. Submitted to Journal of Global Optimization, June, 2019

Li, J., Kirubarajan, T., Tharmarasa: Multi-frame tracking based on multi-dimensional assignment algorithm. To be submitted to IEEE Transactions on Aerospace and Electronic Systems. August, 2019

Chapter 2

A novel three dimensional assignment algorithm

2.1 Introduction

In this chapter, we propose a novel convex dual approach to the three dimensional assignment problem. This approach can be thought of as a type of convexification based on the duality theory. For more on convexification, see, for example,[29, 19, 20]. It will be shown that the optimal value of the proposed dual problem is equal to the best value attainable by the Lagrangian relaxation (LR). Such an elegant formulation makes it possible to conduct theoretical analysis on the algorithm. In fact, we prove two strong duality theorems, which state that under certain mild conditions, the dual problem is equivalent to the original assignment problem. In general cases, the optimal value of the dual problem can provide a satisfactory lower bound on the optimal value of the original assignment problem. Moreover, the newly proposed approach can be extended to higher dimensional (> 3) assignment problems.

The chapter is organized as follows. We first formulate the assignment problem and derive a dual representation. Then, the relationship between the dual approach and the LR approach is investigated. Next, we establish two strong duality theorems for the assignment problem. Finally, we consider the complexity of the algorithm and give some simulation results.

2.2 Formulation of the problem and its dual representation

Given a set of coefficients $\{c_{ijk}\}, (i,j,k) \in I_0$, an index set given by

$$I_0 = \{(i,j,k) | i \in \{1,...,N\}, j \in \{1,...,N\}, k \in \{1,...,N\}\}, \tag{2.2.1}$$

we have a linear objective function

$$f(x) = \sum_{(i,j,k)\in I_0} c_{ijk} x_{ijk}, \tag{2.2.2}$$

where $x = (x_{ijk})$.

For brevity, the following vector notations will be adopted in this chapter:

$$x = (x_{111}, x_{211}, ..., x_{N11}, ..., x_{NN1}, ..., x_{NNN}), \tag{2.2.3}$$

$$\mu = (\mu_1, \mu_2, ..., \mu_N), \tag{2.2.4}$$

$$\nu = (\nu_1, \nu_2, ..., \nu_N), \tag{2.2.5}$$

$$\eta = (\eta_1, \eta_2, ..., \eta_N), \tag{2.2.6}$$

$$\mu^* = (\mu_1^*, \mu_2^*, ..., \mu_N^*), \tag{2.2.7}$$

etc.

In the context of radar tracking, the coefficient c_{ijk} represents the cost associated with assigning track i to the jth measurement z_{1j} from sensor 1 and the kth measurement z_{2k} from sensor 2, and $x = (x_{ijk})$ has the following interpretation:

$$x_{ijk} = \begin{cases} 1 & \text{if measurements } z_{1j} \text{ and } z_{2k} \text{ are associated to track } i, \\ 0 & \text{otherwise.} \end{cases}$$

With such an interpretation, the linear function $f(x)$ then represents the total cost of associating the two groups of measurements (with N measurements from each of the two sensors) to the N tracks. In order to minimize the total cost, we have the following 3-D assignment problem:

$$\min_x f(x), \tag{PP}$$

subject to the constraints:

$$x_{ijk} \in \{0,1\} \tag{2.2.8}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{N} x_{ijk} = 1, \forall i \in \{1,...,N\}, \tag{2.2.9}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{N} x_{ijk} = 1, \forall j \in \{1,...,N\}, \tag{2.2.10}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ijk} = 1, \forall k \in \{1,...,N\}. \tag{2.2.11}$$

The constraints given above specify that each track can only be assigned to one measurement from each sensor, and that every measurement can only be assigned to a single track.

Moreover, we will use S_F to represent the set of feasible solutions to the optimization problem (PP), i.e.,

$$S_F = \{x \in \{0,1\}^{N^3} : \ x \text{ satisfies Eqs.((2.2.9), (2.2.10), (2.2.11)}\}. \tag{2.2.12}$$

In addition, $\forall (i,j,k) \in I_0$, define the reduced costs

$$d_{ijk} = c_{ijk} - \mu_i - \nu_j - \eta_k. \tag{2.2.13}$$

In this chapter, we will consider a special type of Lagrangian:

$$\begin{aligned} L(x,\mu,\nu,\eta) = & \sum_{(i,j,k)\in I_0} c_{ijk}x_{ijk} + \sum_{(i,j,k)\in I_0} |d_{ijk}|(x_{ijk}^2 - x_{ijk}) \\ & + \sum_{i=1}^{N}\big(\mu_i(1 - \sum_{j=1}^{N}\sum_{k=1}^{N} x_{ijk})\big) + \sum_{j=1}^{N}\big(\nu_j(1 - \sum_{i=1}^{N}\sum_{k=1}^{N} x_{ijk})\big) \\ & + \sum_{k=1}^{N}\big(\eta_k(1 - \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ijk})\big). \end{aligned} \tag{2.2.14}$$

Substituting Eq. (2.2.13) into the above equation, we get

$$\begin{aligned} L(x,\mu,\nu,\eta) = & \sum_{(i,j,k)\in I_0} \big(|d_{ijk}|x_{ijk}^2 + (d_{ijk} - |d_{ijk}|)x_{ijk}\big) \\ & + \sum_{i=1}^{N}\mu_i + \sum_{j=1}^{N}\nu_j + \sum_{k=1}^{N}\eta_k. \end{aligned} \tag{2.2.15}$$

For $d_{ijk} \neq 0$, we have

$$\begin{aligned} |d_{ijk}|x_{ijk}^2 + (d_{ijk} - |d_{ijk}|)x_{ijk} &= |d_{ijk}|\left(x_{ijk}^2 + \frac{d_{ijk} - |d_{ijk}|}{|d_{ijk}|}x_{ijk}\right) \\ &= |d_{ijk}|\left(x_{ijk} + \frac{d_{ijk} - |d_{ijk}|}{2|d_{ijk}|}\right)^2 - \frac{(d_{ijk} - |d_{ijk}|)^2}{4|d_{ijk}|} \\ &\geq -\frac{(d_{ijk} - |d_{ijk}|)^2}{4|d_{ijk}|}. \end{aligned} \tag{2.2.16}$$

Thus,

$$|d_{ijk}|x_{ijk}^2 + (d_{ijk} - |d_{ijk}|)x_{ijk} \geq \begin{cases} d_{ijk}, \text{ if } d_{ijk} < 0, \\ 0, \text{ if } d_{ijk} \geq 0. \end{cases} \tag{2.2.17}$$

The above observation leads us to define $\hat{x} = (\hat{x}_{ijk})$

$$\hat{x}_{ijk} = \begin{cases} 1, \text{ if } d_{ijk} < 0; \\ 0, \text{ if } d_{ijk} \geq 0. \end{cases} \tag{2.2.18}$$

(**Note:** $\hat{x}$ is actually a function of (μ, ν, η).) Then, we have

$$\begin{aligned} L(\hat{x}, \mu, \nu, \eta) = & \sum_{(i,j,k)\in I_0} \left(|d_{ijk}| \hat{x}_{ijk}^2 + (d_{ijk} - |d_{ijk}|) \hat{x}_{ijk} \right) \\ & + \sum_{i=1}^{N} \mu_i + \sum_{j=1}^{N} \nu_j + \sum_{k=1}^{N} \eta_k \\ = & \sum_{(i,j,k)\in I_0} \left(d_{ijk} \right)^- + \sum_{i=1}^{N} \mu_i + \sum_{j=1}^{N} \nu_j + \sum_{k=1}^{N} \eta_k, \end{aligned} \tag{2.2.19}$$

where the real piece-wise linear concave function $(\cdot)^-$ is defined as:

$$(\theta)^- = \begin{cases} \theta, \text{ if } \theta < 0; \\ 0, \text{ otherwise.} \end{cases} \tag{2.2.20}$$

From Eqs. (2.2.15, 2.2.17, 2.2.19), we immediately get the following weak duality result:

Lemma 2.2.1. *$\forall \mu, \nu, \eta$, $\forall x \in S_F$, we have*

$$f(x) = L(x, \mu, \nu, \eta) \geq L(\hat{x}, \mu, \nu, \eta).$$

For brevity, we denote $L(\hat{x}, \mu, \nu, \eta)$ by $H(\mu, \nu, \eta)$. Thus, by Eq. (2.2.19),

$$H(\mu, \nu, \eta) = \sum_{(i,j,k)\in I_0} \left(d_{ijk}\right)^{-} + \sum_{i=1}^{N} \mu_i + \sum_{j=1}^{N} \nu_j + \sum_{k=1}^{N} \eta_k. \tag{2.2.21}$$

We will consider the following dual programming problem

$$\max_{\mu,\nu,\eta} H(\mu, \nu, \eta), \tag{PD}$$

and show a strong duality result for this problem and the primal problem (PP) under certain conditions.

Remark: $H(\mu, \nu, \eta)$ is a pointwise infimum of a family of affine functions in λ, μ and ν, and hence is a concave function.

Thus, the dual problem (PD) is a convex optimization problem. In fact, it can be transformed into a linear programming (LP) problem (see Section 2.7). Furthermore, the first term in Eq. (2.2.21) can be thought of as a penalty term, which brings in some useful properties to be discussed in Section 2.4.

2.3 A comparison between the dual approach and the Lagrangian relaxation approach

Before proving the duality results, it is interesting to make a comparison between our pure dual approach and the well established LR approach. We show in this section that the optimal value H^* of the problem (PD) is equal to the optimal value of the objective function used by the LR method[15, 33, 37, 11, 38].

As pointed out in [37], multidimensional assignment algorithms [33, 11, 38] are

extensions of the three dimensional assignment algorithm proposed in [15]. In fact, in the case of the three dimensional assignment problem (3AP), the different variations of the LR method proposed in [33, 11, 38] reduce to nearly the same form as that in [15], although they have been independently developed and have differences in a number of implementation aspects for higher dimensional assignment problems. Therefore, these different formulations of the LR algorithm are considered to be essentially the same algorithm in this chapter. When we refer to the LR method later on, it points to all the variations of the method proposed in [15, 33, 11, 38].

Let S_2 denote the set of $\omega = (\omega_{ij}), i = 1, \ldots, N; j = 1, \ldots, N$ satisfying the constraints:

$$\omega_{ij} \in \{0, 1\} \tag{2.3.1}$$

$$\sum_{j=1}^{N} \omega_{ij} = 1, \forall i \in \{1, ..., N\}, \tag{2.3.2}$$

$$\sum_{i=1}^{N} \omega_{ij} = 1, \forall j \in \{1, ..., N\}. \tag{2.3.3}$$

There are several variants of the LR method for multidimensional assignment problems[15, 33, 11, 38]. However, all of them operate by maximizing the following objective function:

$$G(\eta) = \min_{\omega \in S_2} \Big\{ \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \omega_{ij} \Big\} + \sum_{k=1}^{N} \eta_k, \tag{2.3.4}$$

where

$$a_{ij} = \min_{1 \le k \le N} \big(c_{ijk} - \eta_k \big).$$

In the following discussions, the optimal value of the above objective function will

be denoted by

$$J_{LR}^{*} = \max_{\eta} G(\eta) \tag{2.3.5}$$

which is the lower bound of the optimal assignment cost estimated by the LR method.

We first establish a more tractable representation of the LR bound.

Lemma 2.3.1. *There exist $\hat{\mu}, \hat{\nu}, \hat{\eta}$, such that the bound of the LR method can be represented as*

$$\begin{aligned} J_{LR}^{*} = &\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\min_{1\le k\le N}\big(c_{ijk} - \hat{\eta}_k\big) - \hat{\mu}_i - \hat{\nu}_j\big)^{-} \\ &+ \sum_{i=1}^{N}\hat{\mu}_i + \sum_{j=1}^{N}\hat{\nu}_j + \sum_{k=1}^{N}\hat{\eta}_k. \end{aligned} \tag{2.3.6}$$

Proof. By the continuity of $G(\eta)$, there exists $\hat{\eta} = (\hat{\eta}_k)_{k=1}^{N}$, such that

$$\begin{aligned} J_{LR}^{*} &= G(\hat{\eta}) \\ &= \Big(\min_{\omega\in S_2}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{a}_{ij}\omega_{ij}\Big) + \sum_{k=1}^{N}\hat{\eta}_k \end{aligned} \tag{2.3.7}$$

where

$$\hat{a}_{ij} = \min_{1\le k\le N}\big(c_{ijk} - \hat{\eta}_k\big).$$

Furthermore, it follows from the duality theory for the 2-D assignment problem that

$$\begin{aligned} &\min_{\omega\in S_2}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{a}_{ij}\omega_{ij} \\ &= \max_{\mu,\nu} H_2(\mu,\nu) \end{aligned} \tag{2.3.8}$$

where

$$H_2(\mu,\nu) = \max_{\mu,\nu}\Big\{\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\hat{a}_{ij}-\mu_i-\nu_j\big)^{-}+\sum_{i=1}^{N}\mu_i+\sum_{j=1}^{N}\nu_j\Big\}$$

By the continuity of $H_2(\mu,\nu)$, there exist $\hat{\mu}$ and $\hat{\nu}$, such that

$$\begin{aligned}&\min_{\omega\in S_2}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{a}_{ij}\omega_{ij}\\ &=\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\hat{a}_{ij}-\hat{\mu}_i-\hat{\nu}_j\big)^{-}+\sum_{i=1}^{N}\hat{\mu}_i+\sum_{j=1}^{N}\hat{\nu}_j\end{aligned}$$

Combining the above equation with Eq. (2.3.7), we get

$$\begin{aligned}J^*_{LR} = &\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\min_{1\le k\le N}\big(c_{ijk}-\hat{\eta}_k\big)-\hat{\mu}_i-\hat{\nu}_j\big)^{-}\\ &+\sum_{i=1}^{N}\hat{\mu}_i+\sum_{j=1}^{N}\hat{\nu}_j+\sum_{k=1}^{N}\hat{\eta}_k\end{aligned}\tag{2.3.9}$$

□

□

We also need the following substitution lemma:

Lemma 2.3.2. *There exist $\hat{\mu},\hat{\nu}$ and $\hat{\eta}$ which satisfy Eq. (2.3.6), and furthermore, for each index-pair (i,j), there is at most one index k with*

$$c_{ijk}-\hat{\eta}_k-\hat{\mu}_i-\hat{\nu}_j<0.$$

Proof. We have established the existence of $\hat{\mu},\hat{\nu}$ and $\hat{\eta}$ satisfying Eq. (2.3.6) in the previous lemma. We show now that the claim of the lemma is true for any given

index-pair (i_1, j_1). Let

$$k_1 = \arg \min_{1 \le k \le N} \big(c_{i_1 j_1 k} - \hat{\eta}_k\big). \tag{2.3.10}$$

If

$$c_{i_1 j_1 k_1} - \hat{\eta}_{k_1} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} = 0$$

then, by the definition of k_1, we have

$$c_{i_1 j_1 k} - \hat{\eta}_k - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} \ge 0$$

$\forall k$, and the proof is done.

On the other hand, if

$$c_{i_1 j_1 k_1} - \hat{\eta}_{k_1} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} < 0,$$

and suppose that there is another index k_2 such that

$$c_{i_1 j_1 k_2} - \hat{\eta}_{k_2} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} < 0,$$

then, it follows from the definition of k_1 that

$$c_{i_1 j_1 k_1} - \hat{\eta}_{k_1} \le c_{i_1 j_1 k_2} - \hat{\eta}_{k_2} \tag{2.3.11}$$

Define $\tilde{\mu} = (\tilde{\mu_i})$ by

$$\tilde{\mu_i} = \begin{cases} \hat{\mu}_i - \epsilon, \text{ if } i = i_1, \\ \hat{\mu}_i, \text{ else.} \end{cases} \tag{2.3.12}$$

where

$$\epsilon = -\big(c_{i_1 j_1 k_2} - \hat{\eta}_{k_2} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1}\big) > 0, \tag{2.3.13}$$

then one can verify that

$$c_{i_1 j_1 k_1} - \hat{\eta}_{k_1} - \tilde{\mu}_{i_1} - \hat{\nu}_{j_1} \leq 0 \tag{2.3.14}$$

$$c_{i_1 j_1 k_2} - \hat{\eta}_{k_2} - \tilde{\mu}_{i_1} - \hat{\nu}_{j_1} = 0. \tag{2.3.15}$$

Moreover, by the dual theory of 2-D assignment,

$$\min_{1 \leq k \leq N} \big(c_{i_1 j k} - \hat{\eta}_k\big) - \hat{\mu}_{i_1} - \hat{\nu}_j \geq 0 \tag{2.3.16}$$

$\forall j \neq j_1$.

As a result, we have

$$\Big(\min_{1 \leq k \leq N} \big(c_{i_1 j k} - \hat{\eta}_k\big) - \tilde{\mu}_{i_1} - \hat{\nu}_j\Big)^- = \Big(\min_{1 \leq k \leq N} \big(c_{i_1 j k} - \hat{\eta}_k\big) - \hat{\mu}_{i_1} - \hat{\nu}_j + \epsilon\Big)^- = 0$$

and

$$\Big(\min_{1 \leq k \leq N} \big(c_{i_1 j k} - \hat{\eta}_k\big) - \hat{\mu}_{i_1} - \hat{\nu}_j\Big)^- = 0$$

$\forall j \neq j_1$.

With the above two equations, and by canceling all the $i \neq i_1$ terms, we get

$$\begin{aligned} J^*_{LR} &- \Big\{\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\min_{1\le k\le N}\big(c_{ijk}-\hat{\eta}_k\big)-\tilde{\mu}_i-\hat{\nu}_j\Big)^- \\ &+\sum_{i=1}^{N}\tilde{\mu}_i+\sum_{j=1}^{N}\hat{\nu}_j+\sum_{k=1}^{N}\hat{\eta}_k\Big\} \\ &=\Big(\min_{1\le k\le N}\big(c_{i_1j_1k}-\hat{\eta}_k\big)-\hat{\mu}_{i_1}-\hat{\nu}_{j_1}\Big)^- +\hat{\mu}_{i_1} \\ &-\Big(\min_{1\le k\le N}\big(c_{i_1j_1k}-\hat{\eta}_k\big)-\tilde{\mu}_{i_1}-\hat{\nu}_{j_1}\Big)^- -\tilde{\mu}_{i_1}. \end{aligned} \tag{2.3.17}$$

Therefore,

$$\begin{aligned} J^*_{LR} &- \Big\{\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\min_{1\le k\le N}\big(c_{ijk}-\hat{\eta}_k\big)-\tilde{\mu}_i-\hat{\nu}_j\Big)^- \\ &+\sum_{i=1}^{N}\tilde{\mu}_i+\sum_{j=1}^{N}\hat{\nu}_j+\sum_{k=1}^{N}\hat{\eta}_k\Big\} \\ &=\epsilon+\Big(\big(c_{i_1j_1k_1}-\hat{\eta}_{k_1}\big)-\hat{\mu}_{i_1}-\hat{\nu}_{j_1}\Big)^- \\ &-\Big(\big(c_{i_1j_1k_1}-\hat{\eta}_{k_1}\big)-\tilde{\mu}_{i_1}-\hat{\nu}_{j_1}\Big)^- \\ &=\epsilon+\Big(\big(c_{i_1j_1k_1}-\hat{\eta}_{k_1}\big)-\hat{\mu}_{i_1}-\hat{\nu}_{j_1}\Big) \\ &-\Big(\big(c_{i_1j_1k_1}-\hat{\eta}_{k_1}\big)-\tilde{\mu}_{i_1}-\hat{\nu}_{j_1}\Big) \\ &=0, \end{aligned} \tag{2.3.18}$$

or equivalently,

$$\begin{aligned} J^*_{LR} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\min_{1\le k\le N}\big(c_{ijk}-\hat{\eta}_k\big)-\tilde{\mu}_i-\hat{\nu}_j\Big)^- \\ &+\sum_{i=1}^{N}\tilde{\mu}_i+\sum_{j=1}^{N}\hat{\nu}_j+\sum_{k=1}^{N}\hat{\eta}_k \end{aligned} \tag{2.3.19}$$

Therefore, by replacing the original $\hat{\mu}$ with $\tilde{\mu}$, one can see that the new $\hat{\mu}$, together with $\hat{\nu}$ and $\hat{\eta}$, satisfies Eq. (2.3.6). Furthermore, we now have

$$c_{i_1 j_1 k_1} - \hat{\eta}_{k_1} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} \leq 0 \tag{2.3.20}$$

$$c_{i_1 j_1 k_2} - \hat{\eta}_{k_2} - \hat{\mu}_{i_1} - \hat{\nu}_{j_1} = 0. \tag{2.3.21}$$

By repeating the above procedure for a certain finite number of times, we will be able to find the values of $\hat{\mu}$, $\hat{\nu}$ and $\hat{\eta}$ that have the desired properties. □

□

Lemma 2.3.3. *If (μ^*, ν^*, η^*) is an optimal solution to problem (PD), that is,*

$$\begin{aligned} H^* = &\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\left(c_{ijk} - \eta_k^* - \mu_i^* - \nu_j^*\right)^- \\ &+ \sum_{i=1}^{N}\mu_i^* + \sum_{j=1}^{N}\nu_j^* + \sum_{k=1}^{N}\eta_k^*, \end{aligned} \tag{2.3.22}$$

then, for each pair of index (i, j), there is at most an index k with

$$c_{ijk} - \eta_k^* - \mu_i^* - \nu_j^* < 0.$$

Proof. We show the lemma by a contradiction argument. Suppose that there is an index (i_1, j_1), for which there are two indices k_1 and k_2 such that

$$c_{i_1 j_1 k_1} - \eta_{k_1}^* - \mu_{i_1}^* - \nu_{j_1}^* \leq c_{ijk_2} - \eta_{k_2}^* - \mu_{i_1}^* - \nu_{j_1}^* < 0. \tag{2.3.23}$$

Define $\tilde{\mu} = (\tilde{\mu}_i)$ by

$$\tilde{\mu}_i = \begin{cases} \mu_i^* - \epsilon, \text{ if } i = i_1, \\ \mu_i^*, \text{ else.} \end{cases} \tag{2.3.24}$$

where

$$\epsilon = -\frac{1}{2}\big(c_{ijk_2} - \eta_{k_2}^* - \mu_{i_1}^* - \nu_{j_1}^*\big) > 0, \tag{2.3.25}$$

then one can verify that

$$\big(c_{i_1 j_1 k_1} - \eta_{k_1}^* - \tilde{\mu}_{i_1} - \nu_{j_1}^*\big)^- = \epsilon + \big(c_{i_1 j_1 k_1} - \eta_{k_1}^* - \mu_{i_1}^* - \nu_{j_1}^*\big)^- \tag{2.3.26}$$

$$\big(c_{i_1 j_1 k_2} - \eta_{k_2}^* - \tilde{\mu}_{i_1} - \nu_{j_1}^*\big)^- = \epsilon + \big(c_{i_1 j_1 k_2} - \eta_{k_2}^* - \mu_{i_1}^* - \nu_{j_1}^*\big)^- \tag{2.3.27}$$

and

$$\big(c_{i_1 j_1 k} - \eta_k^* - \tilde{\mu}_{i_1} - \nu_{j_1}^*\big)^- \geq \big(c_{i_1 j_1 k} - \eta_k^* - \mu_{i_1}^* - \nu_{j_1}^*\big)^- \tag{2.3.28}$$

$\forall k$. As a result, we get

$$\begin{aligned} H^* &= \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \big(\big(c_{ijk} - \eta_k^* - \mu_i^* - \nu_j^*\big)^- \\ &\quad + \sum_{i=1}^N \mu_i^* + \sum_{j=1}^N \nu_j^* + \sum_{k=1}^N \eta_k^* \\ &\leq -\epsilon + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \big(\big(c_{ijk} - \eta_k^* - \tilde{\mu}_i - \nu_j^*\big)^- \\ &\quad + \sum_{i=1}^N \tilde{\mu}_i + \sum_{j=1}^N \nu_j^* + \sum_{k=1}^N \eta_k^* \end{aligned} \tag{2.3.29}$$

a contradiction to that H^* is the optimal value of the dual problem (PD). □

□

Now, we can prove the following comparison theorem:

Theorem 2.3.4. *The optimal value H^* of the problem (PD) is equal to the bound J^*_{LR} of the relaxation method, i.e.,*

$$H^* = J^*_{LR} \tag{2.3.30}$$

Proof. Step 1: By Lemma 2.3.2, there exist $\hat{\mu}, \hat{\nu}, \hat{\eta}$, such that

$$\begin{aligned} J^*_{LR} = &\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\min_{1\le k\le N}\big(c_{ijk} - \hat{\eta}_k\big) - \hat{\mu}_i - \hat{\nu}_j\big)^- \\ &+ \sum_{i=1}^{N}\hat{\mu}_i + \sum_{j=1}^{N}\hat{\nu}_j + \sum_{k=1}^{N}\hat{\eta}_k, \end{aligned} \tag{2.3.31}$$

and for each index-pair (i, j), there is at most one index k such that

$$c_{ijk} - \hat{\eta}_k - \hat{\mu}_i - \hat{\nu}_j < 0. \tag{2.3.32}$$

As a result, we get

$$\begin{aligned} J^*_{LR} = &\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\big(c_{ijk} - \hat{\eta}_k - \hat{\mu}_i - \hat{\nu}_j\big)^- \\ &+ \sum_{i=1}^{N}\hat{\mu}_i + \sum_{j=1}^{N}\hat{\nu}_j + \sum_{k=1}^{N}\hat{\eta}_k \end{aligned} \tag{2.3.33}$$

,

On the other hand, the optimal value of the problem (PD) is given by

$$H^* = \max_{\mu,\nu,\eta} \Big\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \big(\big(c_{ijk} - \eta_k - \mu_i - \nu_j \big)^- + \sum_{i=1}^{N} \mu_i + \sum_{j=1}^{N} \nu_j + \sum_{k=1}^{N} \eta_k \Big\}, \tag{2.3.34}$$

by the definition of H in the previous section.

Combining the above two equations, we get

$$H^* \geq J^*_{LR}. \tag{2.3.35}$$

Step 2: Suppose

$$H^* = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \big(c_{ijk} - \eta^*_k - \mu^*_i - \nu^*_j \big)^- + \sum_{i=1}^{N} \mu^*_i + \sum_{j=1}^{N} \nu^*_j + \sum_{k=1}^{N} \eta^*_k. \tag{2.3.36}$$

Then, by Lemma 2.3.3, for each pair of index (i, j), there is at most an index k with

$$c_{ijk} - \eta^*_k - \mu^*_i - \nu^*_j < 0. \tag{2.3.37}$$

Therefore, we get

$$\begin{aligned} H^* &= \sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\min_{1\le k\le N}\big(c_{ijk}-\eta_k^*\big)-\mu_i^*-\nu_j^*\Big)^- \\ &\quad + \sum_{i=1}^{N}\mu_i^* + \sum_{j=1}^{N}\nu_j^* + \sum_{k=1}^{N}\eta_k^* \\ &\le \max_{\mu,\nu}\Big\{\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\min_{1\le k\le N}\big(c_{ijk}-\eta_k^*\big)-\mu_i-\nu_j\Big)^- \\ &\quad + \sum_{i=1}^{N}\mu_i + \sum_{j=1}^{N}\nu_j\Big\} + \sum_{k=1}^{N}\eta_k^* \\ &= \min_{\omega\in S_2}\Big\{\sum_{i=1}^{N}\sum_{j=1}^{N}\big[\min_{1\le k\le N}\big(c_{ijk}-\eta_k^*\big)\big]\omega_{ij}\Big\} + \sum_{k=1}^{N}\eta_k^* \\ &= G(\eta^*) \le J_{LR}^*. \end{aligned} \tag{2.3.38}$$

where we have used the dual theory of 2-D assignment problem, and $G(\eta)$ is defined in Eq. (2.3.4).

Combining the inequalities (2.3.35, 2.3.38) leads to the conclusion of the theorem.

□

□

Remark: All of the variants of the LR method[15, 33, 11, 38] for the three-dimensional assignment problem are based on maximizing the function $G(\eta)$ given by Eq. (2.3.4). The distinctions between them are in the different ways of searching for the feasible assignments. Thus, by Theorem 2.3.4, we conclude that H^* is at least as good as the bounds given by the LR method.

2.4 Strong duality results of the assignment problem

Given (μ^*, ν^*, η^*), we have a set of coefficients

$$d^*_{ijk} = c_{ijk} - \mu^*_i - \nu^*_j - \eta^*_k \tag{2.4.1}$$

$\forall (i, j, k) \in I_0$, by Eq.(2.2.13).

The purpose of this section is to prove two strong duality results.

Theorem 2.4.1. *Let f^* and H^* be the optimal values of the assignment problem (PP) and its dual problem (PD) respectively. Then*

$$f^* = H^*, \tag{2.4.2}$$

if and only if there exists an optimal solution (μ^*, ν^*, η^*) to the dual problem (PD), such that

$$d^*_{i_m j_m k_m} \leq 0, m = 1, \ldots, N, \tag{2.4.3}$$

where each of the three sequences $\{i_1, \ldots, i_N\}$, $\{j_1, \ldots, j_N\}$ and $\{k_1, \ldots, k_N\}$ is a permutation of $\{1, \ldots, N\}$.

The next theorem gives an easier to check condition for the strong duality to be true.

Theorem 2.4.2. *Suppose that there exists an optimal solution (μ^*, ν^*, η^*) to the dual*

problem (PD), such that $d^*_{ijk} \neq 0$ *for* $\forall (i,j,k) \in I_0$*, then*

$$f^* = H^*, \tag{2.4.4}$$

where f^* *and* H^* *are the optimal values of the assignment problem (PP) and its dual problem (PD), respectively.*

If the condition in either of the theorems is satisfied, the dual approach will give the optimal solution to the primal assignment problem. In 2-D assignment problems, similar conditions are guaranteed to be true by the unimodularity of the constraint matrix. In the three dimensional case, we can no longer rely on the unimodularity criterion. Thus, the theoretical implication of the condition needs further investigation. However, this is not too restrictive a condition. It has been shown in our Monte Carlo simulations that even when the condition is not met, the average relative error $\frac{f^*-H^*}{f^*}$ is in the range of 2-5% (see Table 2.6.4). In practical applications, where the cost coefficients typically have simpler structures due to the sparsity of the cost coefficients, the average errors are even smaller, as confirmed in [33, 11].

In order to prove the two theorems, we need to investigate the solutions to the dual problem (PD).

2.4.1 Properties of optimal solutions to the problem (PD)

We will use the following index sets to characterize the optimal solutions to the dual problem (PD).

Definition 1. *For any $i \in \{1, 2, ..., N\}$, define the two-dimensional index set*

$$N_1(i) = \{(j, k) : \ d^*_{ijk} < 0\}. \tag{2.4.5}$$

For any $j \in \{1, 2, ..., N\}$, define the two-dimensional index set

$$N_2(j) = \{(i, k) : \ d^*_{ijk} < 0\}. \tag{2.4.6}$$

For any $k \in \{1, 2, ..., N\}$, define the two-dimensional index set

$$N_3(k) = \{(i, j) : \ d^*_{ijk} < 0\}. \tag{2.4.7}$$

Moreover, we denote the number of elements in a set S by $\text{card}(S)$.

Lemma 2.4.3. *$\forall(\mu^*, \nu^*, \eta^*)$, there exists a $\epsilon_0 > 0$, such that*

$$\begin{aligned} & H(\mu, \nu, \eta) - H(\mu^*, \nu^*, \eta^*) \\ &= \sum_{(i,j,k)\in I_0: d^*_{ijk}<0} \big(\mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\big) \\ &+ \sum_{(i,j,k)\in I_0: d^*_{ijk}=0} \Big(\mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\Big)^- \\ &+ \sum_{i=1}^{N} \big(\mu_i - \mu^*_i + \nu_i - \nu^*_i + \eta_i - \eta^*_i\big) \end{aligned} \tag{2.4.8}$$

$\forall(\mu, \nu, \eta)$ satisfying

$$\sum_{k=1}^{N} \big(|\mu_k - \mu^*_k| + |\nu_k - \nu^*_k| + |\eta_k - \eta^*_k|\big) < \epsilon_0. \tag{2.4.9}$$

Proof. Define

$$\epsilon_0 = \begin{cases} \frac{1}{2}, \text{ if } d^*_{ijk} = 0 \text{ for } \forall (i,j,k) \in I_0, \\ \frac{1}{2}\min\{|d^*_{ijk}| : \ d^*_{ijk} \neq 0\}, \text{ otherwise.} \end{cases} \tag{2.4.10}$$

We directly calculate

$$\begin{aligned} & H(\mu,\nu,\eta) - H(\mu^*,\nu^*,\eta^*) \\ &= \sum_{(i,j,k)\in I_0} \Big(\big(c_{ijk} - \mu_i - \nu_j - \eta_k\big)^- - \big(c_{ijk} - \mu^*_i - \nu^*_j - \eta^*_k\big)^- \Big) \\ &+ \sum_{k=1}^{N} \big(\mu_k - \mu^*_k + \nu_k - \nu^*_k + \eta_k - \eta^*_k\big) \\ &= \sum_{(i,j,k)\in I_0} \Big(\big(d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\big)^- - (d^*_{ijk})^- \Big) \\ &+ \sum_{k=1}^{N} \big(\mu_k - \mu^*_k + \nu_k - \nu^*_k + \eta_k - \eta^*_k\big). \end{aligned}$$

Thus,

$$\begin{aligned} & H(\mu,\nu,\eta) - H(\mu^*,\nu^*,\eta^*) \\ &= \sum_{(i,j,k)\in I_0 : d^*_{ijk} < 0} \Big(\big(d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\big)^- - (d^*_{ijk})^- \Big) \\ &+ \sum_{(i,j,k)\in I_0 : d^*_{ijk} = 0} \big(\mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\big)^- \\ &+ \sum_{k=1}^{N} \big(\mu_k - \mu^*_k + \nu_k - \nu_k * + \eta_k - \eta^*_k\big), \end{aligned}$$

due to the fact that if $d^*_{ijk} > 0$, then $d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k > 0$ by inequality (2.4.9) and Eq. (2.4.10), and thus both $\big(d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\big)^-$ and $\big(d^*_{ijk}\big)^-$ are zero.

For $d^*_{ijk} < 0$, we have

$$\begin{aligned}
&\left(d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\right)^- - \left(d^*_{ijk}\right)^- \\
&= d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k - d^*_{ijk} \\
&= \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k,
\end{aligned}$$

because

$$d^*_{ijk} + \mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k < 0,$$

by inequality (2.4.9) and Eq. (2.4.10) once again.

It follows that

$$\begin{aligned}
&H(\mu, \nu, \eta) - H(\mu^*, \nu^*, \eta^*) \\
&= \sum_{(i,j,k) \in I_0 : d^*_{ijk} < 0} \left(\mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\right) \\
&+ \sum_{(i,j,k) \in I_0 : d^*_{ijk} = 0} \left(\mu^*_i - \mu_i + \nu^*_j - \nu_j + \eta^*_k - \eta_k\right)^- \\
&+ \sum_{i=1}^{N} \left(\mu_i - \mu^*_i + \nu_i - \nu^*_i + \eta_i - \eta^*_i\right).
\end{aligned}$$

□

□

Lemma 2.4.4. *(Optimal solution criterion) Let (μ^*, ν^*, η^*) be an optimal solution to*

the problem (PD). Then, the following inequalities must be satisfied:

$$\begin{aligned}\text{card}(N_1(i)) &\leq 1,\\ \text{card}(N_2(j)) &\leq 1,\\ \text{card}(N_3(k)) &\leq 1,\end{aligned}$$

$\forall (i,j,k) \in I_0$.

Proof. Let $\delta_0 = \epsilon_0$, as defined in Lemma 2.4.3.

Fix $i_0 \in \{1, 2, \ldots, N\}$. Let

$$\mu_i = \begin{cases} \mu^*_{i_0} - \delta, \text{ for } i = i_0, \\ \mu^*_i, \text{ otherwise }, \end{cases} \tag{2.4.11}$$

where $\delta \in (0, \delta_0)$, and $\nu_j = \nu^*_j$, $\eta_k = \eta^*_k$ for all j, k.

Then, by Lemma 2.4.3, we have

$$\begin{aligned} & H(\mu,\nu,\eta) - H(\mu^*,\nu^*,\eta^*) \\ &= \delta\Big(\sum_{(i_0,j,k):d^*_{i_0jk}<0} 1\Big) + \Big(\sum_{(i_0,j,k):d^*_{i_0jk}=0} (\delta)^-\Big) - \delta \\ &= \delta\Big(\sum_{(i_0,j,k):d^*_{i_0jk}<0} 1\Big) - \delta \\ &= \delta\big(\text{card}(N_1(i_0)) - 1\big). \end{aligned} \tag{2.4.12}$$

Suppose that (μ^*, ν^*, η^*) is an optimal solution to the problem (PD).

It follows from Eq. (2.4.12) that for any $i_0 \in \{1, 2, \ldots, N\}$, and any $\delta \in (0, \delta_0)$,

we have

$$\delta\big(\text{card}(N_1(i_0) - 1\big) = H(\mu^* + \delta \cdot e_{i_0}, \nu^*, \eta^*) - H(\mu^*, \nu^*, \eta^*) \leq 0.$$

This immediately leads to

$$\text{card}(N_1(i_0)) \leq 1.$$

In the same manner, we can prove

$$\text{card}(N_2(j)) \leq 1,$$

$$\text{card}(N_3(k)) \leq 1,$$

$\forall j$ and $k \in \{1, 2, \ldots, N\}$.

□

□

2.4.2 Proof of the duality theorems

First, we prove Theorem 2.4.1.

Proof. By Eq. (2.2.13), we have

$$c_{ijk} = d_{ijk} + \mu_i + \nu_j + \eta_k, \tag{2.4.13}$$

$\forall (i, j, k) \in I_0$.

It follows that $\forall x = (x_{ijk}) \in S_F$, we have

$$\begin{aligned} f(x) &= \sum_{(i,j,k)\in I_0} c_{ijk}x_{ijk} \\ &= \sum_{(i,j,k)\in I_0} d_{ijk}x_{ijk} + \sum_{(i,j,k)\in I_0} (\mu_i + \nu_j + \eta_k)x_{ijk} \\ &= \sum_{(i,j,k)\in I_0} d_{ijk}x_{ijk} + \sum_{i=1}^{N}\big(\mu_i + \nu_i + \eta_i\big) \end{aligned} \tag{2.4.14}$$

Suppose that there exists an optimal solution (μ^*, ν^*, η^*) to the dual problem (PD) satisfying the condition of the theorem. Define

$$x^*_{ijk} = \begin{cases} 1, \text{ if } (i,j,k) = (i_m, j_m, k_m), m = 1, \ldots, N; \\ 0, \text{ otherwise.} \end{cases} \tag{2.4.15}$$

Then, $x^* = (x^*_{ijk})$ is a feasible solution to (PP). By Eq. (2.4.14),

$$\begin{aligned} f(x^*) &= \sum_{(i,j,k)\in I_0} d^*_{ijk}x^*_{ijk} + \sum_{i=1}^{N}\big(\mu^*_i + \nu^*_i + \eta^*_i\big) \\ &= \sum_{m=1}^{N} d^*_{i_m j_m k_m} + \sum_{i=1}^{N}\big(\mu^*_i + \nu^*_i + \eta^*_i\big) \\ &= \sum_{m=1}^{N}(d^*_{i_m j_m k_m})^- + \sum_{i=1}^{N}\big(\mu^*_i + \nu^*_i + \eta^*_i\big). \end{aligned} \tag{2.4.16}$$

By Lemma 2.4.4, for each index i, there is at most an index-pair (j,k) with

$d^*_{ijk} < 0$. Thus, it follows from Eq. (2.4.16) that

$$\begin{aligned} f(x^*) &= \sum_{(i,j,k)\in I_0} (d^*_{ijk})^- + \sum_{i=1}^{N} \left(\mu^*_i + \nu^*_i + \eta^*_i\right) \\ &= H(\mu^*, \nu^*, \eta^*). \end{aligned} \tag{2.4.17}$$

The above equation, together the weak duality lemma 2.2.1, leads to $f^* = H^*$, and the proof for the sufficiency is complete.

Next, we proceed to show the necessity of the condition. For that purpose, suppose that $f^* = f(x^*) = H^* = H(\mu^*, \nu^*, \eta^*)$ is true.

Then, there are three sequences $\{i_1, \ldots, i_N\}$, $\{j_1, \ldots, j_N\}$ and $\{k_1, \ldots, k_N\}$, with each of them being a permutation of $\{1, \ldots, N\}$, such that $x^* = (x^*_{ijk})$ can be represented in the form of Eq. (2.4.15).

Now, note that we always have

$$f(x^*) - H(\mu^*, \nu^*, \eta^*) = \sum_{m=1}^{N} d^*_{i_m j_m k_m} - \sum_{(i,j,k)\in I_0} (d^*_{ijk})^- \geq 0. \tag{2.4.18}$$

In order to have $f^* - H^* = 0$, we must have

$$d^*_{i_m j_m k_m} \leq 0,$$

for $m = 1, \ldots, N$. The proof is completed. □

□

Next, we prove Theorem 2.4.2.

Proof. Define

$$x^*_{ijk} = \begin{cases} 1, \text{ if } d^*_{ijk} < 0; \\ 0, \text{ if } d^*_{ijk} > 0. \end{cases} \tag{2.4.19}$$

Then, by Lemma 2.4.4, $x^* = (x^*_{ijk})$ is a feasible solution to the problem (PP). As a result, there are three sequences $\{i_1, \ldots, i_N\}$, $\{j_1, \ldots, j_N\}$ and $\{k_1, \ldots, k_N\}$, and each of them is a permutation of $\{1, \ldots, N\}$, such that $x^* = (x^*_{ijk})$ can be represented in the form of Eq. (2.4.15). Furthermore, we have

$$d^*_{i_m j_m k_m} < 0,$$

for $m = 1, \ldots, N$. Then, the theorem follows from Theorem 2.4.1, and the proof is completed.

□

□

2.5 The solution to the original assignment problem

If the conditions of Theorem 2.4.1 or Theorem 2.4.2 are satisfied, then the optimal value of the problem (PP) is equal to the optimal value of the dual problem. Meanwhile, if Theorem 2.4.1 can be applied, the optimal solution $x^* = (x^*_{ijk})$ to the original

assignment problem is given by

$$x^*_{ijk} = \begin{cases} 1, \text{ if } (i,j,k) = (i_m, j_m, k_m), m = 1, \ldots, N; \\ 0, \text{ otherwise.} \end{cases} \tag{2.5.1}$$

If the condition of Theorem 2.4.2 are satisfied, the solution is

$$x^*_{ijk} = \begin{cases} 1, \text{ if } d^*_{ijk} < 0; \\ 0, \text{ if } d^*_{ijk} > 0. \end{cases} \tag{2.5.2}$$

On the other hand, if the conditions of the theorems are not satisfied, we need to search for the final (sometimes sub-optimal) assignment. In principle, we can apply any heuristic searching method [22],[39] on the reduced costs $\{d^*_{ijk}\}$. To get the final assignment in our simulations, however, we used the following simple two-step procedure, which is actually used by the Lagrangian method in each of its iterations to search for a feasible solution [15, 33, 11, 38]:

Step 1: We solve the relaxed 2-D assignment problem:

$$\min_{\omega \in S_2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \omega_{ij} \tag{2.5.3}$$

where S_2 is defined in Section 2.3, and

$$a_{ij} = \min_{1 \le k \le N} \left(c_{ijk} - \eta^*_k \right), \tag{2.5.4}$$

where η^* has already been determined in solving problem (PD).

Step 2: Fix the assignment $(i, j(i))$ obtained in the previous step, and solve another

2-D assignment problem:

$$\min_{\omega \in S_2} \sum_{i=1}^{N} \sum_{k=1}^{N} c_{ij(i)k} \omega_{ik} \tag{2.5.5}$$

to obtain the 3-D assignment $(i, j(i), k(i))$.

2.6 Simulation results

In this section, two sets of simulations are performed to demonstrate the effectiveness of the proposed dual approach. Our computation platform is a laptop with an Intel i7-6500U CPU running Windows 10. The computations are carried out using the MATLAB programming language.

In the first set of simulations, we illustrate the performance of the newly proposed approach by conducting Monte Carlo simulations on some small-size problems. The costs are randomly generated, and hence we get dense three dimensional cost arrays. The algorithms are run 200 times in the simulations.

The true optimal values to problem (PP) are shown in Fig. 2.6.1 for problem size $N = 4$ and in Fig. 2.6.2 for $N = 6$, together with the lower bound estimates H^* by solving the dual problem (PD). To make it easier to compare the results, the differences between the true optimal values and the estimates are also shown in the figures. From the curves, it can be seen that for $N = 4$ the duality gaps are nearly zero in more than half of the simulated problems. Furthermore, it has been shown by calculation that the average relative errors between the estimate H^* and the optimal assignment costs are around 2–3% (see Table 2.6.1).

Next, we demonstrate that the solutions by the Lagrange relaxation method converge towards the optimal solution to the dual problem (PD) with the increasing

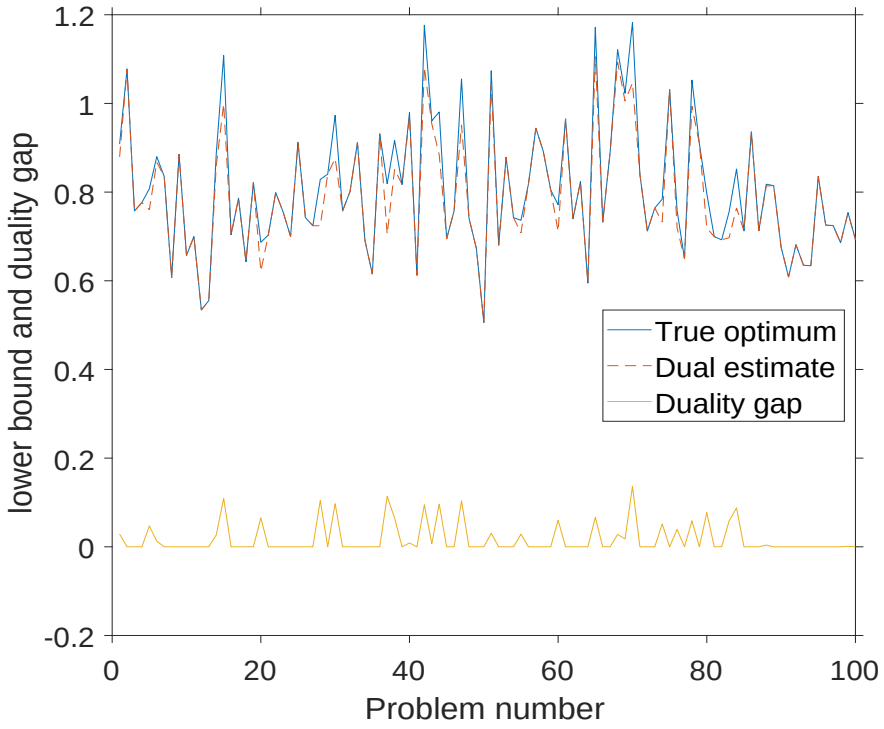

Figure 2.6.1: Lower bound estimates by the dual method ($N = 4$); the curve near zero shows the duality gaps (always non-negative).

Size	Relative error (%)
4x4x4	1.870
6x6x6	2.95

Table 2.6.1 Average errors for $N = 4$ and $N = 6$.

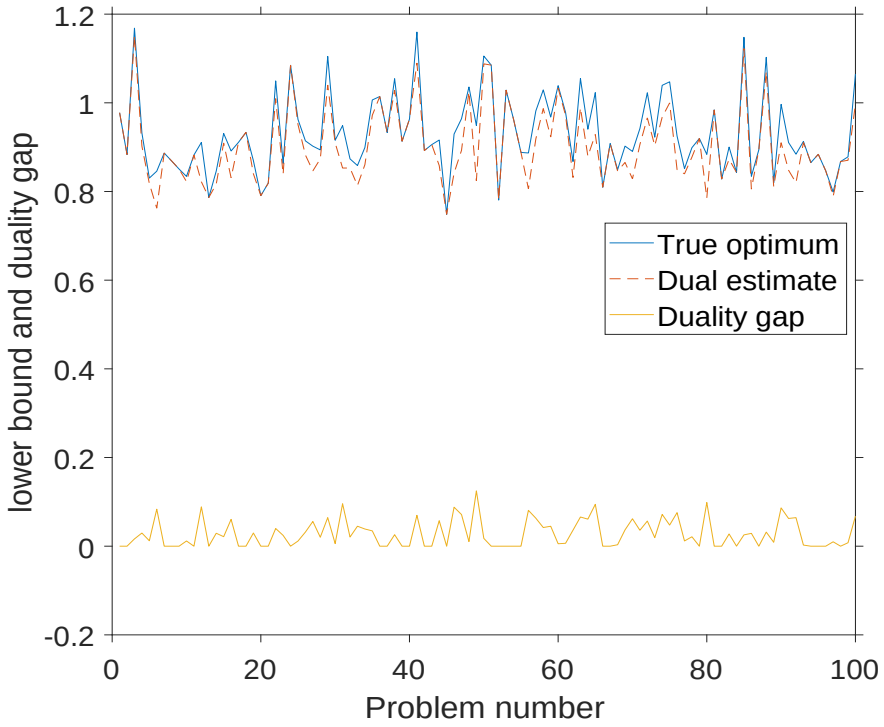

Figure 2.6.2: Lower bound estimates by the dual method ($N = 6$); the curve near zero shows the duality gaps (always non-negative).

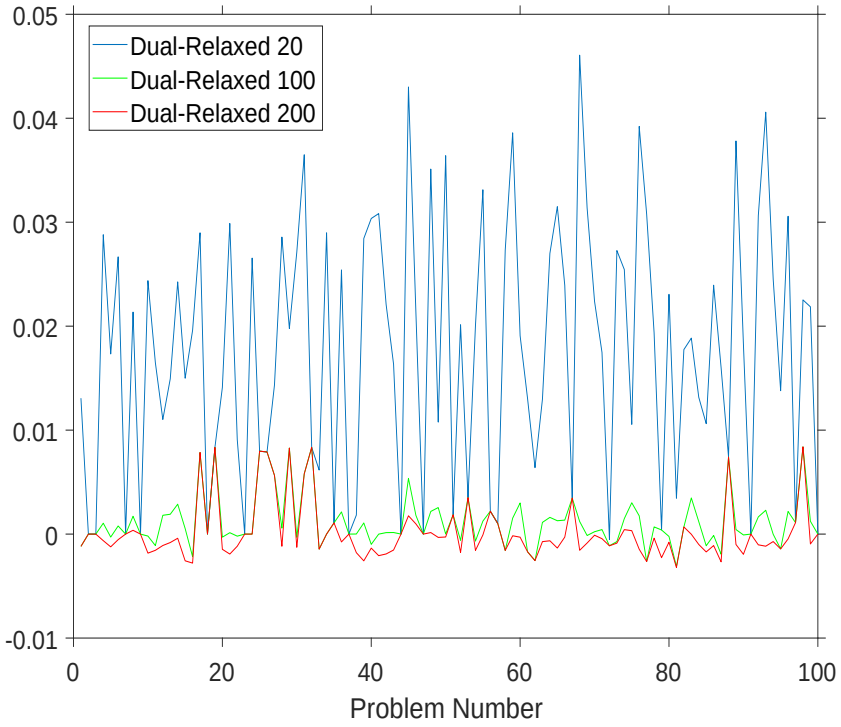

Figure 2.6.3: Differences between the dual estimates and the estimates by the LR method (size N = 6). Top: after 20 LR iterations; Middle: after 100 LR iterations; Bottom: after 200 LR iterations. The optimal values are around 1, similar to those shown in Fig. 2.6.2.

number of iterations. As shown in Fig. 2.6.3, after 100 iterations, the results of the LR method are close to the corresponding results of the dual method, as predicted by Theorem 2.3.4. In fact, due to numerical errors, some points on the bottom curve are even below zero.

In the second set of simulations, we compare the dual approach to the LR method [15, 33, 11, 38] by applying the algorithms to larger size problems. For that purpose, the method proposed in [18] is employed to generate random assignment problems of different sizes with known optimal solutions. The generated problems have dense cost coefficient arrays. Although this is not true for problems in typical tracking applications [33, 11, 38], the randomly generated problems can be used to assess the performance of general assignment algorithms. The cost coefficients are integers in the range $(0, 20)$. Problems of ten different sizes are considered, from small 10x10x10 problems to relatively large 100x100x100 problems. The choice of the sizes is based on[25], oriented towards tracking applications.

The statistics of the randomly generated assignment problems, together with that of the differences between the assignment results, are shown in Table 2.6.2 and Table 2.6.3. Moreover, Figs. (2.6.4–2.6.9) show the distributions of the 40x40x40 and 50x50x50 assignment problems using histograms, together with the distributions of the resulting assignment values for the two classes of problems.

The subgradient method [42] is used to solve the maximization problems produced by both the dual method and the LR method [33, 11]. Before applying the subgradient method, the cost coefficients are normalized so that the normalized cost values are in

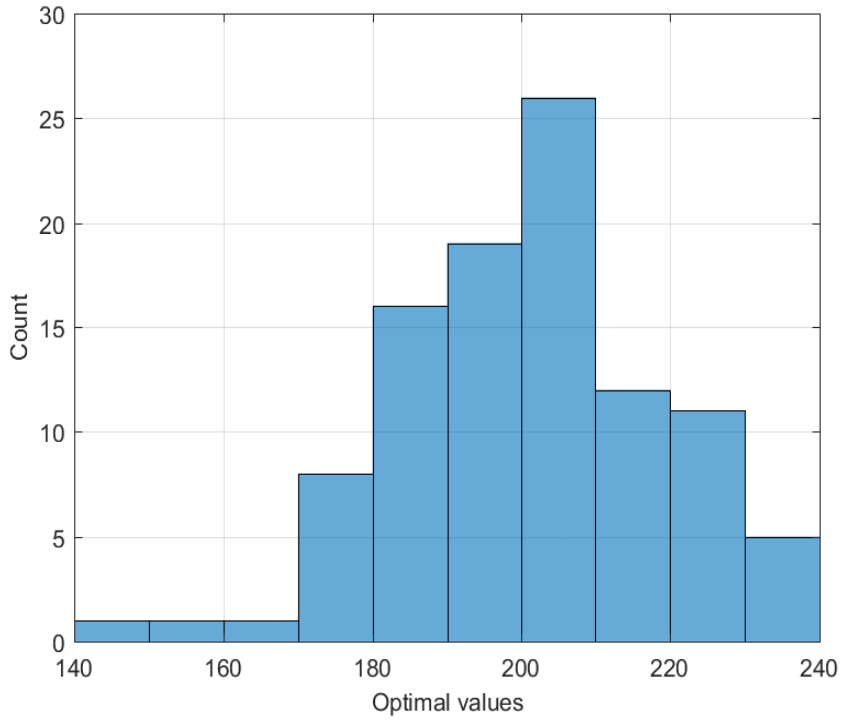

Figure 2.6.4: Distribution of the optimal values of the 100 randomly generated 40x40x40 test problems.

Problem size	Optimal value	Difference 1	Difference 2
10x10x10	48.7000	-0.6812	1.6000
20x20x20	100.3700	-2.0025	3.9300
30x30x30	147.7300	-2.5872	6.4500
40x40x40	200.5400	-3.5150	10.4100
50x50x50	252.2000	-4.3150	10.7800
60x60x60	300.5800	-4.2553	11.6300
70x70x70	342.8250	-5.6898	11.3000
80x80x80	401.5500	-7.9812	9.7000
90x90x90	451.8500	-10.9125	10.9500
100x100x100	492	-15.6250	10.7000

Table 2.6.2 Means of the optimal assignment costs (column 2), the differences between the dual lower bounds and the optimal values (column 3) and the differences between the actual assignment costs and the optimal values (column 4) for different problem sizes.

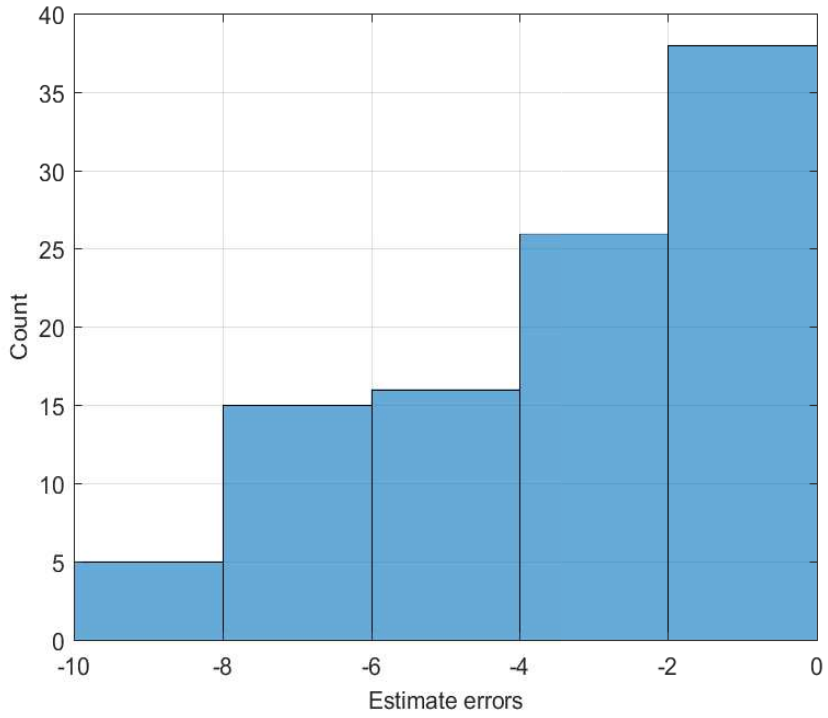

Figure 2.6.5: Distribution of the differences between the dual lower bounds and the optimal values of the 100 randomly generated 40x40x40 test problems.

Problem size	Optimal value	Difference 1	Difference 2
10x10x10	78.6566	0.8816	7.1313
20x20x20	140.1142	3.1506	12.9748
30x30x30	215.8759	4.3009	15.6439
40x40x40	308.5539	5.7585	4.4262
50x50x50	341.1111	5.3575	6.5370
60x60x60	295.6804	6.4773	6.2557
70x70x70	471.8404	6.8194	3.7538
80x80x80	699.6289	5.0957	9.8000
90x90x90	561.0816	6.2093	2.7868
100x100x100	411.7778	8.2281	6.2333

Table 2.6.3 Variances of the optimal assignment costs (column 2), variances of the differences between dual lower bounds and the optimal values (column 3) and variances of the differences between actual assignment costs and optimal values (column 4) for different problem sizes.

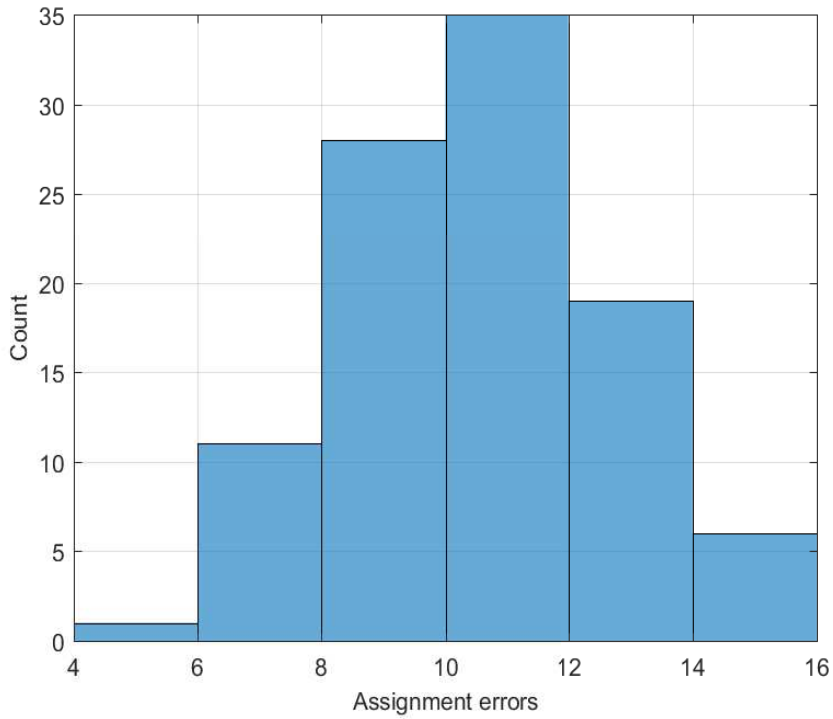

Figure 2.6.6: Distribution of the differences between the assignment results and the optimal values of the 100 randomly generated 40x40x40 test problems.

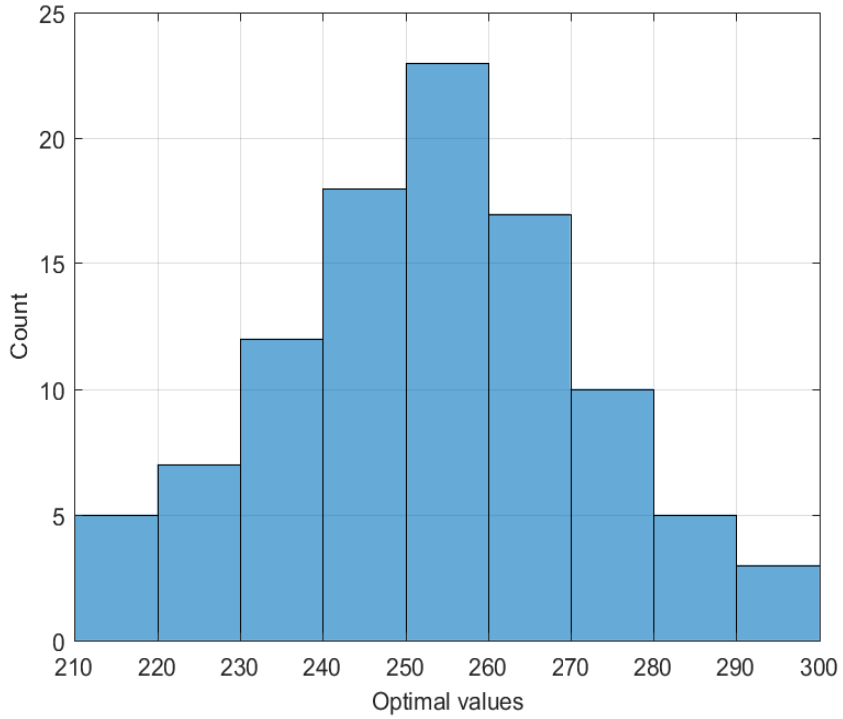

Figure 2.6.7: Distribution of the optimal values of the 100 randomly generated 50x50x50 test problems.

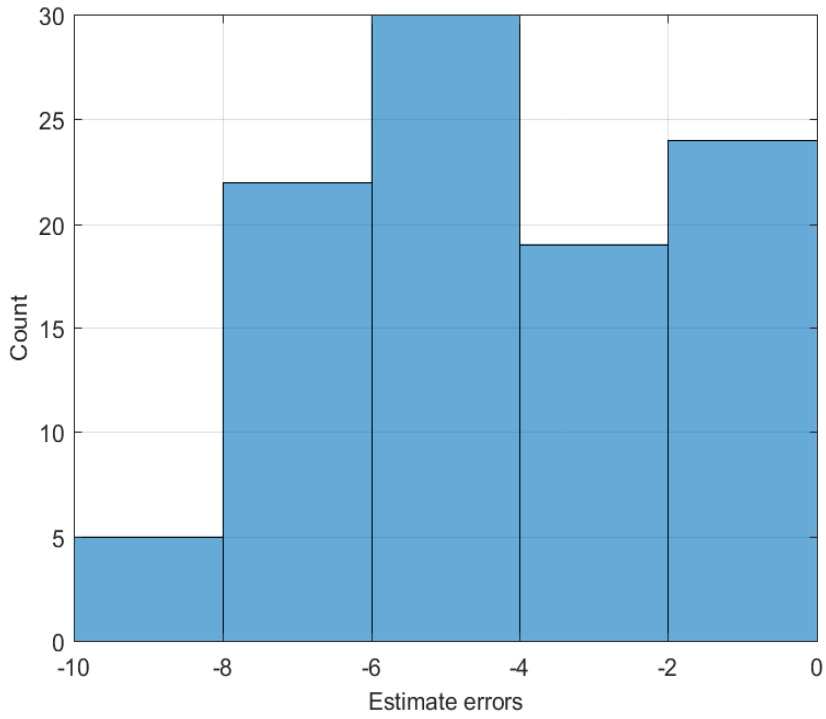

Figure 2.6.8: Distribution of the differences between the dual lower bounds and the optimal values of the 100 randomly generated 50x50x50 test problems.

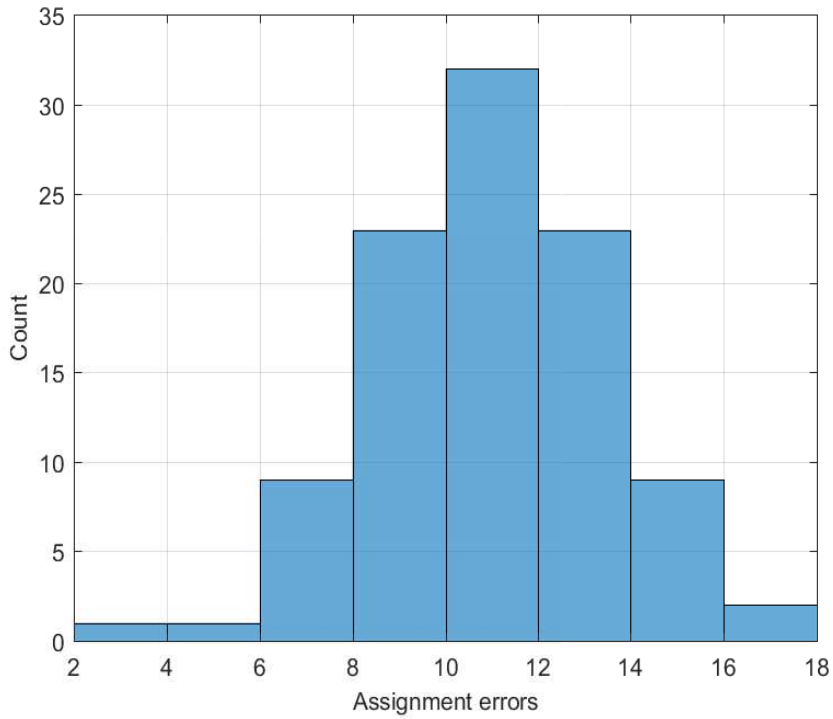

Figure 2.6.9: Distribution of the differences between the assignment results and the optimal values of the 100 randomly generated 50x50x50 test problems.

the range $(0, 1)$. The iteration scheme is

$$x_{n+1} = x_n + \frac{1}{n}\frac{g(x)}{||g(x)||}, \tag{2.6.1}$$

where $g(x) \in \partial f(x)$ with $f(x)$ being the objective function to be maximized, and ∂ denotes the subgradient operator. The reason for choosing such a simple method is to ensure that the differences in the simulation results for the two algorithms under tests are indeed caused by the inherent differences between the dual algorithm and the LR algorithm, instead of being produced by the methods used to solve the associated maximization problems.

For the LR method, the stopping criterion adopted in the simulation is

$$\frac{A_n - G_n}{G_n} < 1\%, \tag{2.6.2}$$

with an upper limit of 200 on the number of iterations, where A_n is the total cost corresponding to the assignment result and G_n is the lower bound estimation at the nth step. The auction algorithm [7] is employed to find the solutions to all resulting 2-D assignment problems in implementing the Lagrange relaxation method.

For the dual method, the stopping criterion is $n > 100$ and

$$|H_{n+1} - H_n| < 0.0001 H_n, \tag{2.6.3}$$

with an upper limit of 200 on the number of iterations, where H_n is the lower bound estimation at the nth step.

The results are shown in Fig. 2.6.11 to Fig. 2.6.20. In these figures, the relative

Problem size	Average relative error(%)	
	Dual	LR
10x10x10	3.223297	4.556535
20x20x20	3.837284	5.215574
30x30x30	4.378541	5.050015
40x40x40	5.226154	5.954631
50x50x50	4.289758	5.000658
60x60x60	3.870937	4.004356
70x70x70	3.312801	3.263973
80x80x80	2.400072	2.816428
90x90x90	2.42686	2.35371
100x100x100	2.1812	2.009

Table 2.6.4 Relative errors for different problem sizes

errors (errors divided by the corresponding optimal values) of the final assignments and the corresponding lower bounds for the test problems are shown. The average relative errors are summarized in Table 2.6.4.

The CPU times for the 40x40x40 simulation are shown in Fig. 2.6.10, with a mean value of 0.2021s and a standard deviation of 0.0220s. Also shown in the figure are the times for the LR method, with a standard variation of 0.0891s. The curves of CPU times for other problem sizes have similar characteristics. The average running times for these tests are summarized in Table 2.6.5.

From the above results, several observations can be made. First, we can see that the dual method outperforms the LR method, both in terms of accuracy and running times.

Second, note that the relative errors of the dual method are better than the relative errors of the LR method, in spite of the fact that both methods have obtained nearly the same lower bounds. This implies that the final assignment results depend not

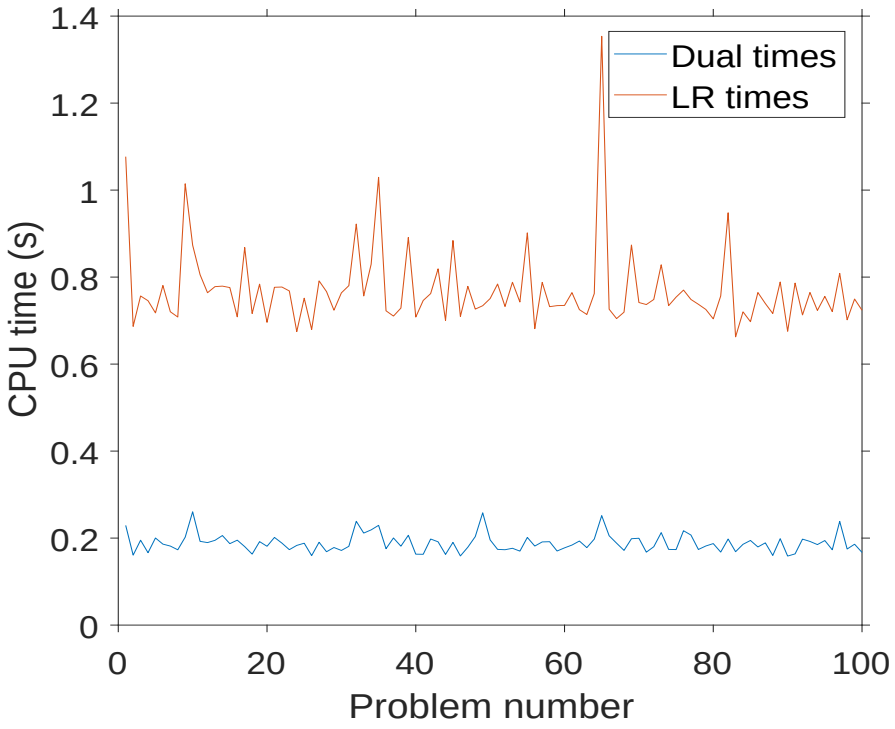

Figure 2.6.10: CPU times for the 100 randomly generated 40x40x40 test problems.

Problem size	CPU time for each problem (s)	
	Dual	LR
10x10x10	0.003223	0.045565
20x20x20	0.020363	0.165529
30x30x30	0.070448	0.457757
40x40x40	0.1921	0.7755
50x50x50	0.718940	2.063451
60x60x60	1.432292	3.662854
70x70x70	2.631925	6.278000
80x80x80	4.057100	8.709200
90x90x90	5.2350	9.5303
100x100x100	6.4780	11.5207

Table 2.6.5 Average elapsed CPU times for different problem sizes

only on the lower bound estimation, but also on the solution (μ, ν, η). The optimal solution to the dual problem (PD) has better symmetry than that obtained by the LR method, in which one set of constraints are relaxed, thus leading to asymmetry among the three sets of constraints. Intuitively, a symmetric solution might capture more complete information about the original assignment problem. Since the solution is not unique, it needs further investigation to determine a good criterion for selecting the solution to the dual problem (PD), or to the problem $\max_\eta G(\eta)$ for the LR method, where $G(\eta)$ is given by Eq. (2.3.4), such that the final assignment result has the smallest error.

Furthermore, when the size of the problem reaches 70x70x70, the average relative errors of the two algorithms get close to each other. This is due to the inefficiency of the subgradient method [41]. The drawbacks of the subgradient method become more serious with larger size problems. The size N^3 of the problem (PD) increases much faster than the size N^2 of the maximization problem associated with the LR method when N increases. As a result, the advantage of the dual method gradually decreases when the size reaches 60x60x60. For problems larger than 100x100x100, more efficient numerical methods are necessary to fully demonstrate the power of the dual method. Alternatives to the subgradient iteration scheme considered here, along with the generalization of the dual solution to higher dimensional assignment problems, is the topic of chapter 2.

On the other hand, the statistics in Table 2.6.2 and Table 2.6.3 and in Figs. (2.6.4–2.6.9) compare the distribution of the original optimal values with that of the errors between the solutions and the optimal values. We see that after applying the dual algorithm, the variances of original distributions have been reduced significantly.

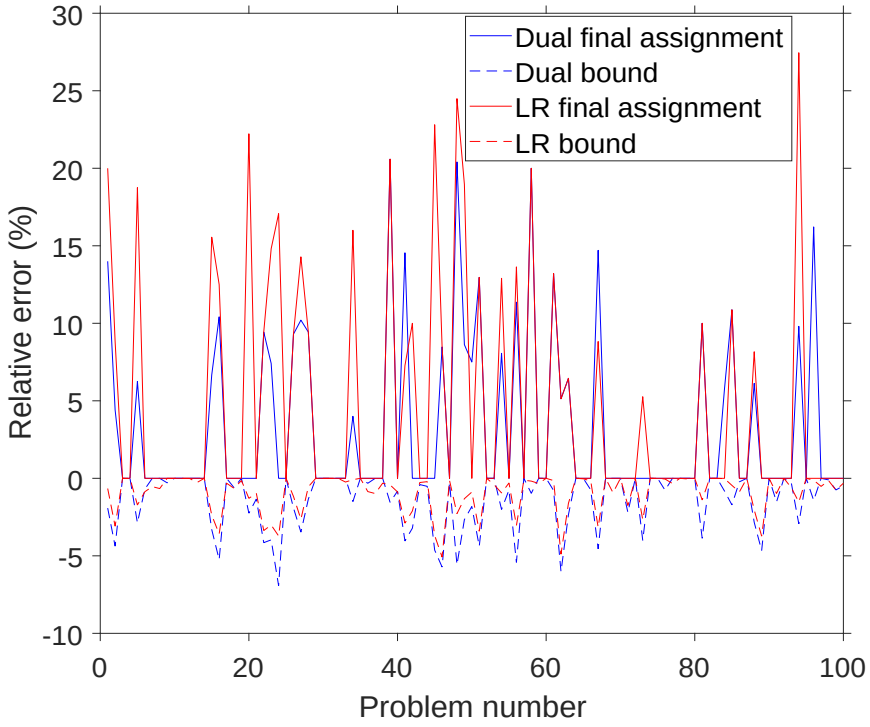

Figure 2.6.11: Assignment and lower bound estimation relative errors for one hundred 10x10x10 randomly generated test problems ($N = 10$).

This demonstrates the power of the algorithm from a statistical perspective.

2.7 Remarks on the time complexity of the dual problem

We have shown that under some mild conditions, three-dimensional assignment problems can be transformed into a mathematically equivalent (in terms of their optimal values) unconstrained convex programming problems through the duality approach. The original problem is known to be NP-hard [16]. Now, let us examine the time

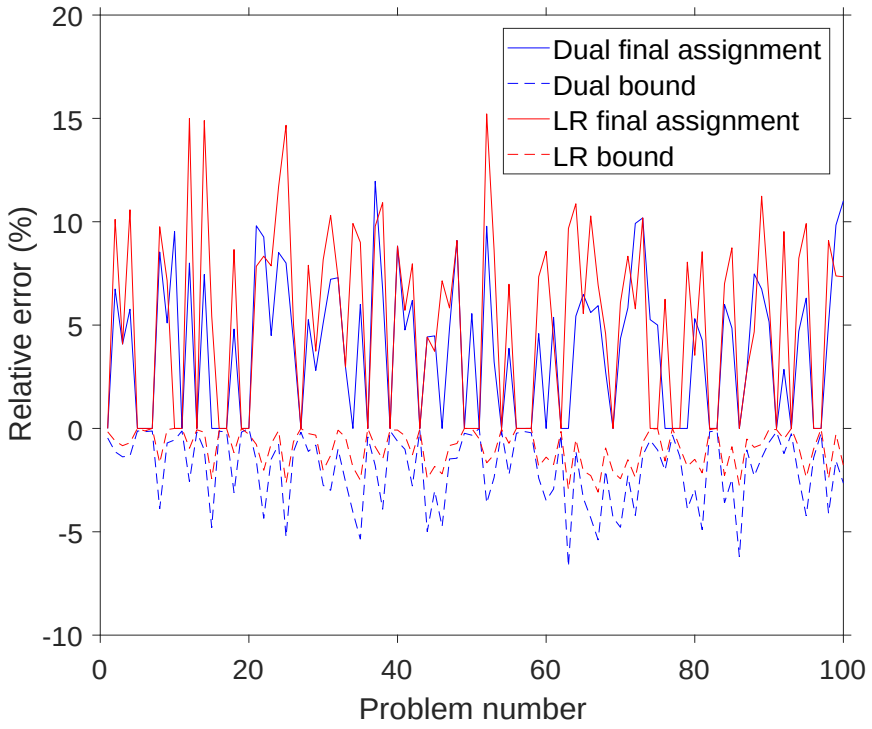

Figure 2.6.12: Assignment and lower bound estimation relative errors for one hundred 20x20x20 randomly generated test problems ($N = 20$).

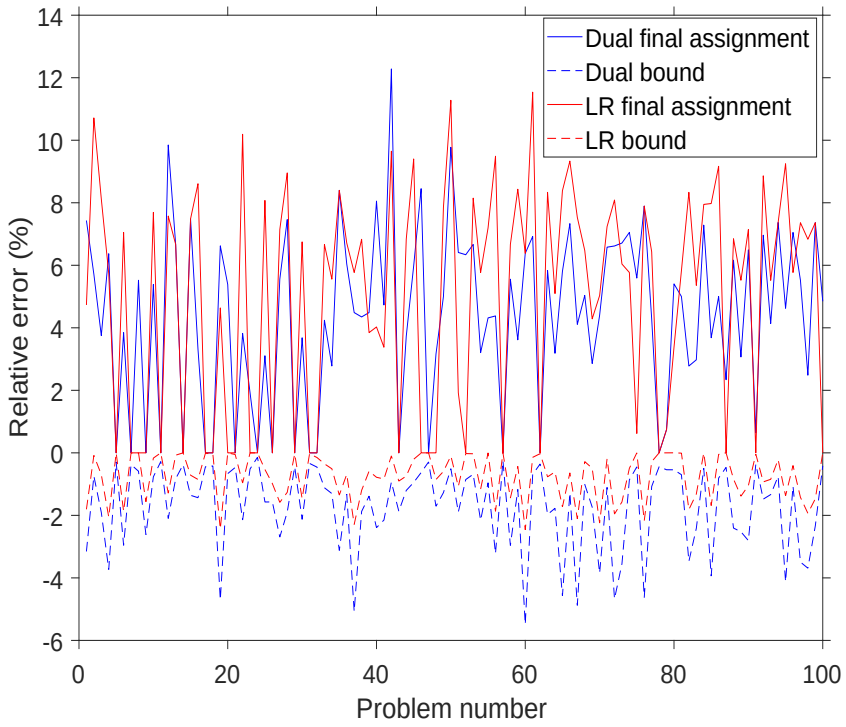

Figure 2.6.13: Assignment result and lower bound estimation relative errors for one hundred 30x30x30 randomly generated test problems ($N = 30$).

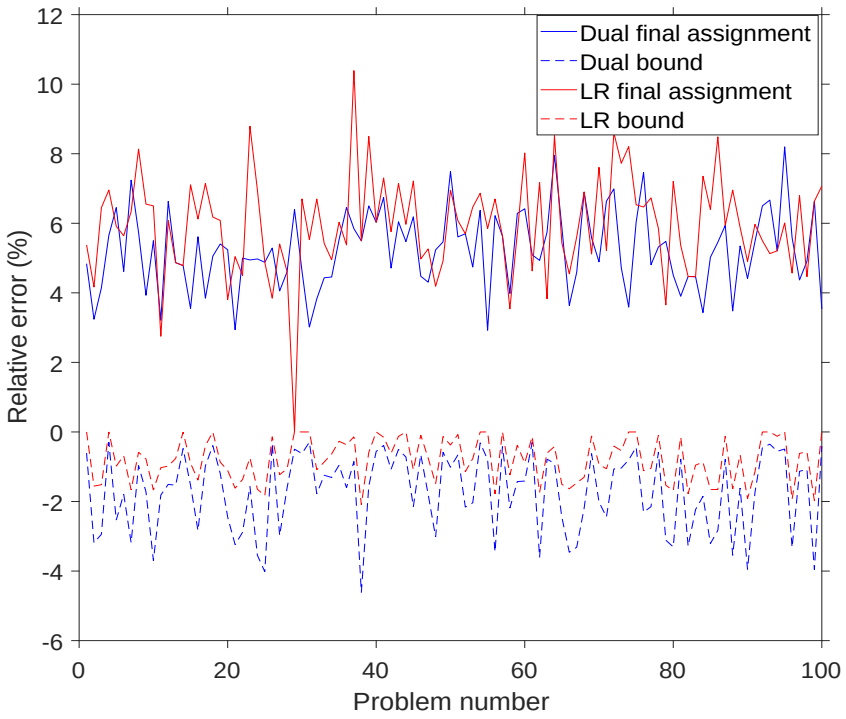

Figure 2.6.14: Assignment and lower bound estimation relative errors for one hundred 40x40x40 randomly generated test problems ($N = 40$).

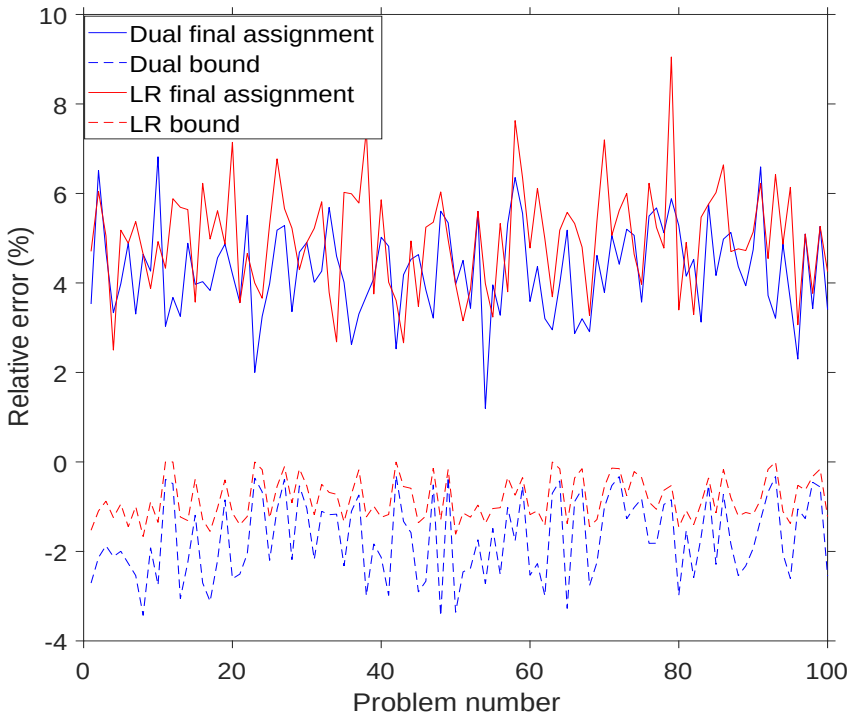

Figure 2.6.15: Assignment and lower bound estimation relative errors for one hundred 50x50x50 randomly generated test problems ($N = 50$).

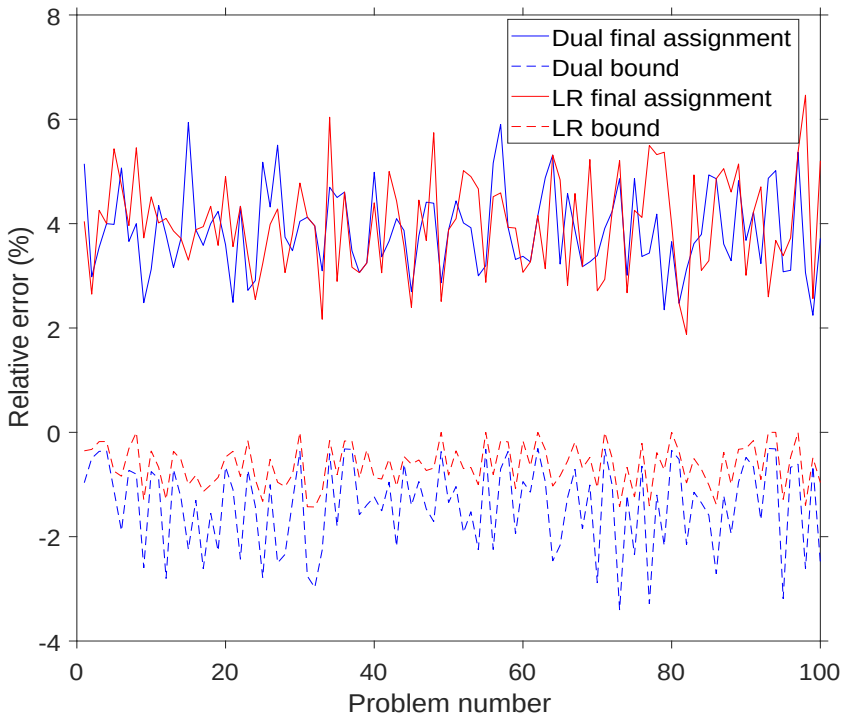

Figure 2.6.16: Assignment and lower bound estimation relative errors for one hundred 60x60x60 randomly generated test problems ($N = 60$).

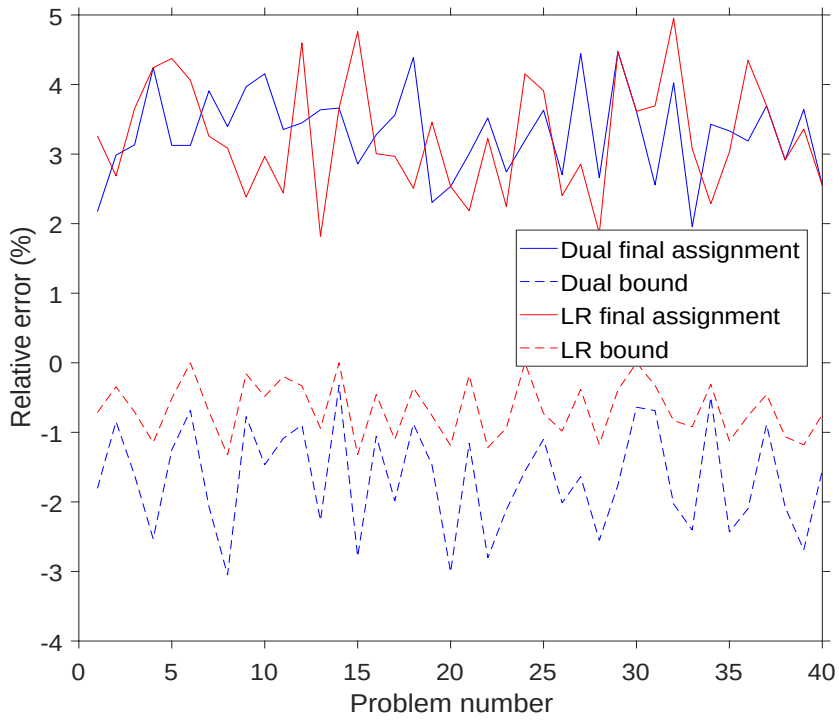

Figure 2.6.17: Assignment and lower bound estimation relative errors for fourty 70x70x70 randomly generated test problems ($N = 70$).

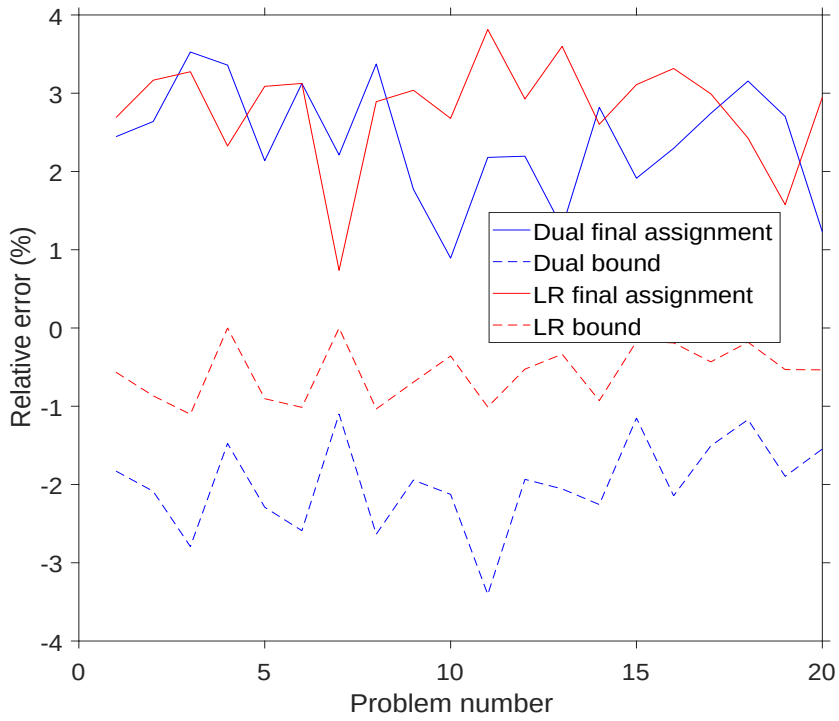

Figure 2.6.18: Assignment and lower bound estimation relative errors for twenty 80x80x80 randomly generated test problems ($N = 80$).

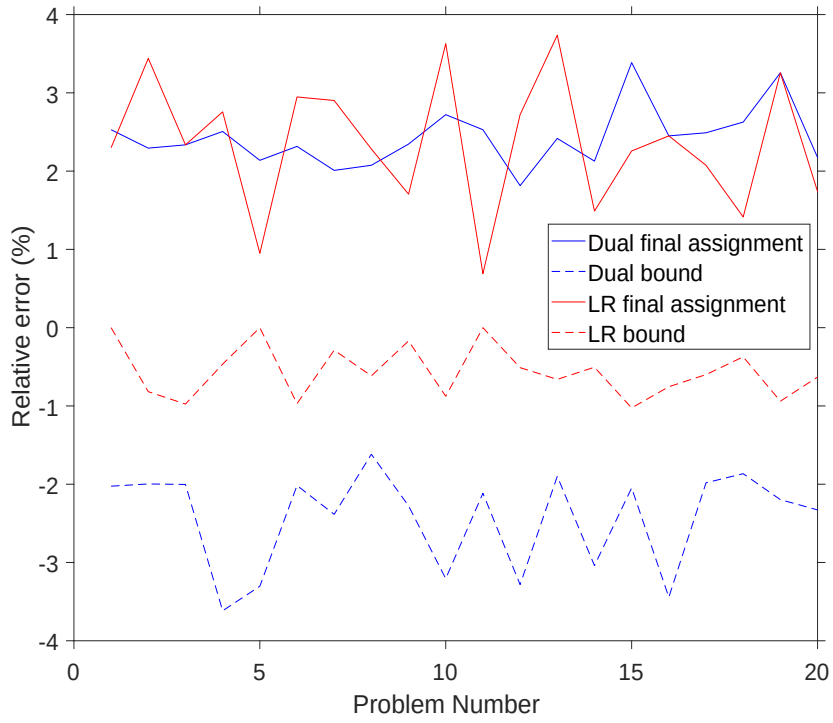

Figure 2.6.19: Assignment and lower bound estimation relative errors for twenty 90x90x90 randomly generated test problems ($N = 90$).

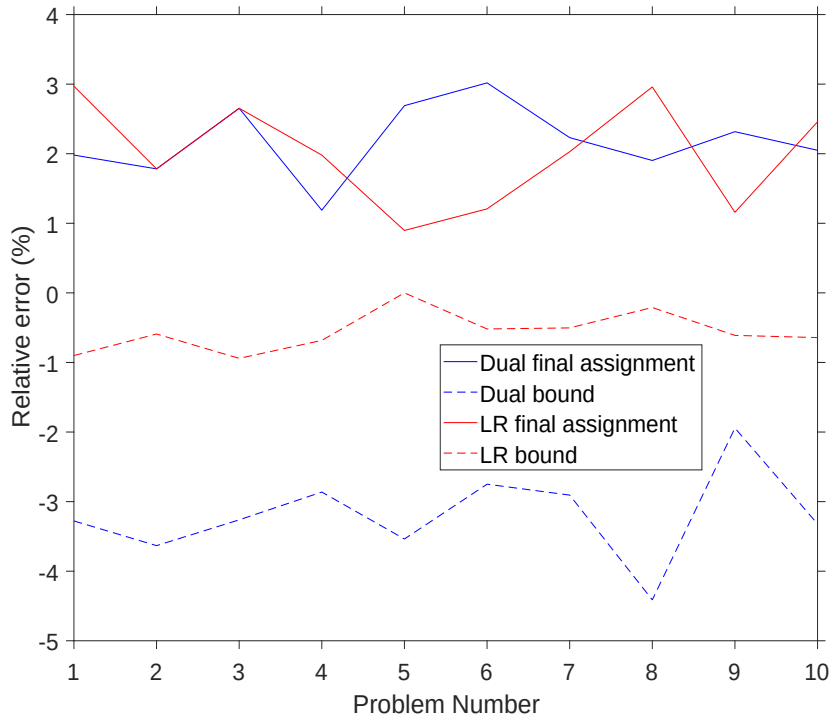

Figure 2.6.20: Assignment and lower bound estimation relative errors for ten 100x100x100 randomly generated test problems ($N = 100$).

complexity of the resulting dual problem (PD), which is equivalent to the following:

$$\min_{\mu,\nu,\eta} h(\mu,\nu,\eta), \tag{PDE}$$

where

$$\begin{aligned} h(\mu,\nu,\eta) &= \frac{1}{2}\Big(\sum_{(i,j,k)\in I_0} c_{ijk} - H(\mu,\nu,\eta)\Big) \\ &= \sum_{(i,j,k)\in I_0} |c_{ijk} - \mu_i - \nu_j - \eta_k| \\ &\quad + (N^2-2)\sum_{i=1}^{N}\mu_i + (N^2-2)\sum_{j=1}^{N}\nu_j \\ &\quad + (N^2-2)\sum_{k=1}^{N}\eta_k. \end{aligned}$$

The problem (PDE) is a convex optimization problem. A variety of iterative algorithms, such as the subgradient algorithm used in our simulation example, are available to compute the optimal solution to (PDE) highly efficiently. With such algorithms, the time complexity depends on the size of the problem, the choice of the initial solution, the iteration step sizes, as well as accuracy of the resulting solution [42, 14, 5]. Typically, the iteration converges to the solution in hundreds or even tens of steps in typical applications [42, 14]. From a practical point of view, the computational complexity of the problem (PDE) will be $\propto N^3$, if we set the number of iteration steps to a fixed number, say, 100.

Furthermore, the problem (PDE) can be represented in an equivalent LP form:

$$\min_{\mu,\nu,\eta,t} \sum_{(i,j,k)\in I_0} t_{ijk} + (N^2-2)\sum_{i=1}^{N}\big(\mu_i+\nu_i+\eta_i\big) \tag{LP}$$

subject to

$$c_{ijk} - \mu_i - \nu_j - \eta_k \leq t_{ijk}, \forall (i,j,k) \in I_0,$$

and

$$c_{ijk} - \mu_i - \nu_j - \eta_k \geq -t_{ijk}, \forall (i,j,k) \in I_0.$$

The problem has $3N + N^3$ variables, and subject to $2N^3$ constraint inequalities.

By Khachiyan's ellipsoid algorithm, the problem (LP) can be solved in polynomial time [23]. Interior methods present another more efficient way to solve LP problems in polynomial time.

The time complexity of the LR method depends on the algorithm used to solve the 2-D assignment problems. When the 2-D problems are solved using the auction method, the time complexity follows $O(N^2)$ [7] but depends on cost coefficients. To simplify the comparison, let us assume the 2-D assignment problem can be solved by the auction algorithm in $N^2 \times K_1$ calculations, where K_1 is a fixed number, say 100. Then the time complexity of the relaxation method is $N^2 \times K_1 \times K_2$, where K_2 is the number of iterations by the subgradient algorithms, say, 200.

The dual problem can be solved in $N^3 \times K_3$ calculations, where K_3 is the number of iterations by the subgradient algorithms, say, 200. Then, we see that if $K_2 = K_3$, then the relative time complexity of the dual method and the LR method depends on the ratio N/K_1. If the size of the assignment problem $N < K_1$, or $N < 100$, assuming $K_1 = 100$, then the dual method has a relatively lower time complexity. On the other hand, if $N > 100$, then the relaxation method will have a lower time complexity than the dual method. For tracking applications, if the number of tracks is less than 100, the dual method should be more efficient than the relaxation method based on the

above reasoning. This was confirmed in our previous simulations.

In our implementation, the time complexity of the relaxation method is close to that of the dual method. However, it should be pointed out that the performance of the dual algorithm can be further improved by employing more efficient numerical methods such as the Gauss-Seidel based iteration method [6]. These implementation related issues, together with other ways of boosting the efficiency of the proposed algorithm, will be discussed in the next chapters.

2.8 Conclusions

In this chapter, a dual approach was proposed to convert three-dimensional assignment problems into unconstrained convex programming problems. It is shown that the optimal value of the dual problem is equal to the upper bound of the objective functions of the relaxation method. Both the dual method and the LR method demonstrate high accuracy around 97% for large size problems. However, the advantages of the dual approach are twofold:

First, it makes theoretical analysis of the algorithm more tractable. In fact, based on the dual representation, two strong duality results were proved.

Second, for small and medium size problems, the dual method is more efficient and has higher accuracy than the LR method even with the basic subgradient method. For large size problems, the problem (PD) can be transformed into a linear optimization problem. This is a key advantage, since it makes it possible to employ highly efficient linear optimization algorithms, such as the interior point method and the classical simplex method, which are usually much faster than nonlinear optimization algorithms, to solve the maximization problem (PD). Considering the vast amount

of literature on linear optimization algorithms, this is indeed a significant advantage, and it will enable the proposed method to solve even larger assignment problems arising in a wide variety of applications.

Another advantage of the dual approach is that it can be extended to solve generalized assignment problems, such as many-to-one and one-to-many assignment problems, and assignment problems with inequality constraints. Furthermore, the method presented in this chapter can be extended to higher dimensional cases.

Chapter 3

Dual algorithm for multi-dimensional assignment problems

3.1 Introduction

In the previous chapter, we developed a novel dual approach for the three-dimensional assignment problem, also see [31]. The dual method demonstrated the same or better accuracy than the Lagrangian relaxation (LR) method while being more time-efficient and easier to implement due to the simple structure of the dual formulation.

In this chapter, we extend the dual approach to solve more general multidimensional assignment problems. The chapter is organized as follows. A dual convex representation is derived first in Section 3.2. Also, we discuss the condition under which the duality gap reaches zero in Section 3.3. Next, we investigate relationship

of the dual approach to the LR method in Section 3.4.Furthermore, a numerical algorithm based on the dual formulation is presented in Section 3.5. The simulation results given in Section 3.6 show that the proposed algorithm outperforms the LR approach both in terms of accuracy and computational efficiency. Finally, we consider in Section 3.7 the time complexity of the algorithm and propose an improved iteration scheme called the alternating direction iteration method.

It should be pointed out that the results presented in this chapter on the Multidimensional assignment problem (MAP) are not a trivial generalization of those given in the previous work [31]. First, the proofs in this chapter are not only simpler and more elegant, but also more powerful in terms of the results than those given in the last chapter. In fact, the proofs given in [31] on the equivalence between the dual formulation and the LR for three dimensional assignment problems cannot be used here to show the equivalence in higher dimensional cases. Second, the new proof also shows that the process of LR is essentially equivalent to relaxing the binary constraint condition in the original assignment problem, a new result that does not appear in the previous work. Moreover, the alternating direction iteration method given in Section 3.7, which converts a high dimensional maximization problem into a set of low dimensional maximization problems, turns out to be essential to overcome the inefficiency of the subgradient method in high dimensional maximization problems. The simulation results show that by such an iteration scheme, the solution process can be significantly accelerated. Finally, an effective search strategy is proposed to further reduce the gap between the final assignment cost and the optimal cost.

3.2 Formulation of the problem and its dual representation

Let $M \geq 3$ be an integer. Given a set of coefficients $\{c_{i_1 \ldots i_M}\}, (i_1, \ldots, i_M) \in I_0$, we have a linear objective function

$$f(x) = \sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1 \ldots i_M} x_{i_1 \ldots 1_M}, \tag{3.2.1}$$

where $x = (x_{i_1 \ldots i_M})$, and I_0 is an index set given by

$$I_0 = \{1, \ldots, N\}^M = \{(i_1, \ldots, i_M) \mid i_1 \in \{1, \ldots, N\}, \ldots, i_M \in \{1, \ldots, N\}\}, \tag{3.2.2}$$

.

The M-dimensional assignment problem can be formulated as

$$\min_x f(x), \tag{PP}$$

subject to the constraints

$$x_{i_1 \ldots i_M} \in \{0, 1\} \tag{3.2.3}$$

$$\sum_{i_2=1}^{N} \cdots \sum_{i_M=1}^{N} x_{i_1 \ldots i_M} = 1, \forall i_1 \in \{1, ..., N\}, \tag{3.2.4}$$

$$\sum_{i_1=1}^{N} \cdots \sum_{i_{m-1}=1}^{N} \sum_{i_{m+1}=1}^{N} \cdots \sum_{x_{i_M}=1}^{N}, \forall i_m \in \{1, ..., N\}, \tag{3.2.5}$$

$$\sum_{i_1=1}^{N} \cdots \sum_{i_{M-1}=1}^{N} x_{i_1 \ldots i_M} = 1, \forall i_M \in \{1, ..., N\}. \tag{3.2.6}$$

In the context of radar tracking applications, $c_{i_1 \ldots i_M}$ can be interpreted as the cost associated with assigning the i_mth measurement z_{m-1,i_m} of sensor $m-1$ ($m = 2, \ldots, M$) to track i_1, and $x = (x_{i_1 \ldots i_M})$ has the following meaning:

$$x_{i_1 \ldots i_M} = \begin{cases} 1 & \text{if measurement } z_{m-1,i_m} \text{ is associated to track } i_1, m = 2, \ldots, M \\ 0 & \text{otherwise.} \end{cases}$$

For brevity, the following vector notations will be adopted in this chapter:

$$x = (x_{11\ldots1}, x_{21\ldots1}, ..., x_{N1\ldots1}, ..., x_{N\ldots N1}, ..., x_{N\ldots NN}),$$

$$\mu_1 = (\mu_{1,1}, \mu_{1,2}, ..., \mu_{1,N}),$$

$$\mu_1^* = (\mu_{1,1}^*, \mu_{1,2}^*, ..., \mu_{1,N}^*),$$

$$...,$$

$$\mu_M = (\mu_{M,1}, \mu_{M,2}, ..., \mu_{M,N}),$$

$$\mu_M^* = (\mu_{M,1}^*, \mu_{M,2}^*, ..., \mu_{M,N}^*),$$

$$\mu = (\mu_1, \mu_2, \ldots, \mu_M).$$

Moreover, we will use S_F to represent the set of feasible solutions to the optimization problem (PP), i.e.,

$$S_F = \{x \in \{0,1\}^{N^M} : \ x \text{ satisfies Eqs.((3.2.4), (3.2.5), (3.2.6)}\}. \tag{3.2.7}$$

In addition, $\forall (i_1, \ldots, i_M) \in I_0$, define the reduced costs

$$d_{i_1 \ldots i_M} = c_{i_1 \ldots i_M} - \mu_{1,i_1} - \mu_{2,i_2} - \cdots - \mu_{M,i_M}. \tag{3.2.8}$$

The development of the dual algorithm is based on a special type of separable Lagrangian:

$$\begin{aligned} L(x,\mu) = & \sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1\ldots i_M} x_{i_1\ldots i_M} + \sum_{(i_1,\ldots,i_M)\in I_0} |d_{i_1\ldots i_M}|(x^2_{i_1\ldots i_M} - x_{i_1\ldots i_M}) \\ & + \sum_{i_1=1}^{N} \big(\mu_{1,i_1}(1 - \sum_{i_2=1}^{N} \cdots \sum_{i_M=1}^{N} x_{i_1\ldots i_M})\big) + \ldots \\ & + \sum_{i_M=1}^{N} \big(\mu_{M,i_M}(1 - \sum_{i_1=1}^{N} \cdots \sum_{i_{M-1}=1}^{N} x_{i_1\ldots i_M})\big). \end{aligned} \tag{3.2.9}$$

Equivalently,

$$\begin{aligned} L(x,\mu) = & \sum_{(i_1,\ldots,i_M)\in I_0} \big(|d_{i_1\ldots i_M}| x^2_{i_1\ldots i_M} + (d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)x_{i_1\ldots i_M}\big) \\ & + \sum_{i_1=1}^{N} \mu_{1,i_1} + \cdots + \sum_{i_M=1}^{N} \mu_{M,i_M}, \end{aligned} \tag{3.2.10}$$

by Eq. (3.2.8).

For $d_{i_1\ldots i_M} \neq 0$, we have

$$\begin{aligned} & |d_{i_1\ldots i_M}| x^2_{i_1\ldots i_M} + (d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)x_{i_1\ldots i_M} \\ =& |d_{i_1\ldots i_M}| \left(x^2_{i_1\ldots i_M} + \frac{d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|}{|d_{i_1\ldots i_M}|} x_{i_1\ldots i_M} \right) \\ =& |d_{i_1\ldots i_M}| \left(x_{i_1\ldots i_M} + \frac{d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|}{2|d_{i_1\ldots i_M}|} \right)^2 - \frac{(d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)^2}{4|d_{i_1\ldots i_M}|} \\ \geq & - \frac{(d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)^2}{4|d_{i_1\ldots i_M}|}. \end{aligned} \tag{3.2.11}$$

Thus,

$$|d_{i_1\ldots i_M}|x^2_{i_1\ldots i_M} + (d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)x_{i_1\ldots i_M} \geq \begin{cases} d_{i_1\ldots i_M}, \text{ if } d_{i_1\ldots i_M} < 0, \\ 0, \text{ if } d_{i_1\ldots i_M} \geq 0. \end{cases} \tag{3.2.12}$$

Defining $\hat{x} = (\hat{x}_{i_1\ldots i_M})$

$$\hat{x}_{i_1\ldots i_M} = \begin{cases} 1, \text{ if } d_{i_1\ldots i_M} < 0; \\ 0, \text{ if } d_{i_1\ldots i_M} \geq 0, \end{cases} \tag{3.2.13}$$

we have the unconstrained Lagrangian

$$\begin{aligned} L(\hat{x}, \mu) = & \sum_{(i_1,\ldots,i_M)\in I_0} \left(|d_{i_1\ldots i_M}|\hat{x}^2_{i_1\ldots i_M} + (d_{i_1\ldots i_M} - |d_{i_1\ldots i_M}|)\hat{x}_{i_1\ldots i_M}\right) \\ & + \sum_{i=1}^{N} \left(\mu_{1,i} + \cdots + \mu_{M,i}\right) \\ = & \sum_{(i_1,\ldots,i_M)\in I_0} \left(d_{i_1\ldots i_M}\right)^- + \sum_{m=1}^{M}\sum_{i_m=1}^{N} \mu_{m,i_m}, \end{aligned} \tag{3.2.14}$$

where the function $(\cdot)^-$ is defined by

$$(\theta)^- = \min(0, \theta) \tag{3.2.15}$$

and is a piece-wise linear concave function.

Note: $\hat{x}$ is a function of $(\mu_1, \ldots, \mu_M)$, see Eq. (3.2.8).

From Eqs. (3.2.10, 3.2.12, 3.2.14), we get the following weak duality result:

Lemma 3.2.1. *$\forall \mu$, $\forall x \in S_F$, we have*

$$f(x) = L(x, \mu) \geq L(\hat{x}, \mu).$$

For brevity, we denote $L(\hat{x}, \mu)$ by $H(\mu)$. Thus, by Eq. (3.2.14),

$$H(\mu) = \sum_{(i_1,\ldots,i_M)\in I_0} \left(d_{i_1\ldots i_M}\right)^- + \sum_{m=1}^{M}\sum_{i_m=1}^{N} \mu_{m,i_m}. \tag{3.2.16}$$

We will consider the following dual problem

$$\max_{\mu} H(\mu), \tag{PD}$$

and show that it leads to an efficient algorithm for the MAP problem.

Remark: $H(\mu)$ is a piece-wise linear concave function. Thus, the dual problem (PD) is a convex optimization problem. In fact, it can be transformed into a linear programming (LP) problem (see Section 3.4).

3.3 Strong duality results for the assignment problem

Given any $(\mu_1^*, \ldots, \mu_M^*)$, we have a set of coefficients

$$d^*_{i_1\ldots i_M} = c_{i_1\ldots i_M} - \mu^*_{1,i_1} - \cdots - \mu^*_{M,i_M} \tag{3.3.1}$$

$\forall (i_1, \ldots, i_M) \in I_0$, by Eq.(3.2.8).

The purpose of this section is to prove two strong duality results.

Theorem 3.3.1. *Let f^* and H^* be the optimal values of the assignment problem (PP) and its dual problem (PD) respectively. Then,*

$$f^* = H^* \tag{3.3.2}$$

if and only if there exists an optimal solution $(\mu_1^*, \ldots, \mu_M^*)$ to the dual problem (PD), such that

$$d^*_{i_{1,n}\cdots\, i_{M,n}} \leq 0, n = 1, \ldots, N, \tag{3.3.3}$$

and for any $m \in \{1, \ldots, M\}$, the sequence $(i_{m,1}, \ldots, i_{m,N})$ is a permutation of $(1, \ldots, N)$,

The next theorem makes it possible to check for the condition for the strong duality.

Theorem 3.3.2. *Suppose that there exists an optimal solution $(\mu_1^*, \ldots, \mu_M^*)$ to the dual problem (PD) such that $d^*_{i_1 \ldots i_M} \neq 0$ for any $(i_1, \ldots, i_M) \in I_0$. Then,*

$$f^* = H^*, \tag{3.3.4}$$

where f^* and H^* are the optimal values of the assignment problem (PP) and its dual problem (PD), respectively.

To prove the two theorems, we need to investigate the solutions to the dual problem (PD).

3.3.1 Properties of optimal solutions to the problem (PD)

We will use the following index sets to characterize the optimal solutions to the dual problem (PD).

Definition 2. *For any* $m \in \{1, \ldots, M\}$, $i \in \{1, 2, ..., N\}$, *define the* $(M-1)$*-dimensional index set*

$$N_m(i_m) = \{(i_1, \ldots, i_{m-1}, i_{m+1}, \ldots, i_M) : \ d^*_{i_1 \ldots i_M} < 0\}. \tag{3.3.5}$$

Moreover, we denote the number of elements in a set S by $\text{card}(S)$.

Lemma 3.3.3. *(Optimal solution criterion) Let* $\mu^* = (\mu^*_1, \ldots, \mu^*_M)$ *be an optimal solution to the problem (PD). Then, the following inequalities must be satisfied:*

$$\text{card}(N_m(i_m)) \leq 1,$$

$\forall m \in \{1, \ldots, M\}, i \in \{1, \ldots, N\}$.

Proof. Suppose that the claim is false. By rearranging the indices, if necessary, we can assume that $d_{111\ldots1} < 0$, and $d_{121\ldots1} < 0$.

Let $\delta = \min\{|d_{1i_2\ldots i-M}| : d_{1i_2,\ldots,i_M} < 0\}$.

Define $\hat{\mu}^* = (\hat{\mu}^*_{m,i_m})$ by

$$\hat{\mu}^*_{m,i_m} = \begin{cases} \mu^*_{m,i_m} - \delta, \ \text{if } m = 1, i_m = 1, \\ \mu^*_{m,i_m}, \ \text{otherwise.} \end{cases} \tag{3.3.6}$$

Then, one can verify that

$$H(\hat{\mu}^*) - H(\mu^*) \geq \delta > 0. \tag{3.3.7}$$

This is however in contradiction to the assumption that μ^* is an optimal solution to the problem (PD). □

□

3.3.2 Proof of the duality theorems

First, we prove Theorem 3.3.1.

Proof. By Eq. (3.2.8), we have

$$c_{i_1 \ldots i_M} = d_{i_1 \ldots i_M} + \mu_{1,i_1} + \cdots + \mu_{M,i_M}, \tag{3.3.8}$$

$\forall (i_1, \ldots, i_M) \in I_0$.

It follows that $\forall x = (x_{i_1 \ldots i_M}) \in S_F$, we have

$$\begin{aligned} f(x) &= \sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1 \ldots i_M} x_{i_1 \ldots i_M} \\ &= \sum_{(i_1,\ldots,i_M)\in I_0} d_{i_1 \ldots i_M} x_{i_1 \ldots i_M} + \sum_{(i_1,\ldots,i_M)\in I_0} (\mu_{1,i_1} + \cdots + \mu_{M,i_M}) x_{i_1 \ldots i_M} \\ &= \sum_{(i_1,\ldots,i_M)\in I_0} d_{i_1 \ldots \ i_M} x_{i_1 \ldots i_M} + \sum_{i=1}^{N} \big(\mu_{1,i} + \cdots + \mu_{M,i}\big) \end{aligned} \tag{3.3.9}$$

Suppose that there exists an optimal solution $(\mu_1^*, \ldots, \mu_M^*)$ to the dual problem

(PD) satisfying the condition of the theorem. Define

$$x^*_{i_1\dots i_M} = \begin{cases} 1, \text{ if } (i_1,\dots,i_M) = (i_{1,n},\dots,i_{M,n}), n = 1,\dots,N; \\ 0, \text{ otherwise.} \end{cases} \tag{3.3.10}$$

Then, $x^* = (x^*_{i_1\dots i_M})$ is a feasible solution to (PP).

Moreover, by Eq. (3.3.9),

$$\begin{aligned} f(x^*) &= \sum_{(i_1,\dots,i_M)\in I_0} d^*_{i_1\dots\ i_M} x^*_{i_1\dots i_M} + \sum_{i=1}^{N}\big(\mu^*_{1,i} + \cdots + \mu^*_{M,i}\big) \\ &= \sum_{n=1}^{N} d^*_{i_{1,n}\dots i_{M,n}} + \sum_{i=1}^{N}\big(\mu^*_{1,i} + \cdots + \mu^*_{M,i}\big) \\ &= \sum_{n=1}^{N}(d^*_{i_{1,n}\dots i_{M,n}})^- + \sum_{i=1}^{N}\big(\mu^*_{1,i} + \cdots + \mu^*_{M,i}\big). \end{aligned} \tag{3.3.11}$$

By Lemma 3.3.3, for each $i_{m,n}$, there is at most one index-tuple $(k_1,\dots,k_M)$ with $i_{m,n} = k_m$, and $d^*_{k_1\dots k_M} < 0$. Thus, it follows from Eq. (3.3.11) that

$$\begin{aligned} f(x^*) &= \sum_{(k_1,\dots,k_M)\in I_0} (d^*_{k_1\dots k_M})^- + \sum_{i=1}^{N}\big(\mu^*_{1,i} + \cdots + \mu^*_{M,i}\big) \\ &= H(\mu^*_1,\dots,\mu^*_M). \end{aligned} \tag{3.3.12}$$

The above equation, together with the weak duality lemma 3.2.1, leads to $f^* = H^*$, and the proof for the sufficiency is complete.

Next, we proceed to show the necessity of the condition. For that purpose, suppose that x^* is an optimal solution to (PP) and $f^* = f(x^*) = H^* = H(\mu^*_1,\dots,\mu^*_M)$ is true.

Then, there are M sequences $\{i_{m,1},\dots,i_{m,N}\}$, $m = 1,\dots,M$, with each of these

sequences being a permutation of $\{1,\ldots,N\}$, such that $x^* = (x^*_{i_1\ldots i_M})$ can be represented in the form of Eq. (3.3.10).

Now, note that

$$f(x^*) - H(\mu_1^*,\ldots,\mu_M^*) = \sum_{n=1}^{N} d^*_{i_{1,n}\ldots i_{M,n}} - \sum_{(i_1,\ldots,i_M)\in I_0} (d^*_{i_1\ldots i_M})^- \geq 0. \tag{3.3.13}$$

In order to satisfy $f^* - H^* = 0$, we must have

$$d^*_{i_{1,n}\ldots i_{M,n}} \leq 0,$$

for $n = 1,\ldots,N$. The proof is completed. □

□

Next, we prove Theorem 3.3.2.

Proof. Define

$$x^*_{i_1\ldots i_M} = \begin{cases} 1, \text{ if } d^*_{i_1\ldots i_M} < 0; \\ 0, \text{ if } d^*_{i_1\ldots i_M} > 0. \end{cases} \tag{3.3.14}$$

Then, by Lemma 3.3.3, $x^* = (x^*_{i_1\ldots i_M})$ is a feasible solution to the problem (PP). As a result, there are M sequences $\{i_{m,1},\ldots,i_{m,N}\}$, where $m = 1,\ldots,M$, with each of these sequences being a permutation of $\{1,\ldots,N\}$, such that $x^* = (x^*_{i_1\ldots i_M})$ can be represented in the form of Eq. (3.3.10). Furthermore, we have

$$d^*_{i_{1,n}\ldots i_{M,n}} < 0,$$

for $n = 1,\ldots,N$. Then, the theorem follows from Theorem 3.3.1, and the proof is

completed.

□

□

3.4 Connection to the Lagrangian relaxation approach

Now, let us investigate the connection between the proposed dual approach and the LR approach. The major result of this section is that the optimal value H^* of the problem (PD) is equal to the optimal value of the objective function maximized by the LR method[15, 33, 11, 38].

There are several variants of the LR method for multidimensional assignment problems[15, 33, 11, 38]. We will focus on one of the variants[33, 11].

Let $S_F^{[m-1]}$ denote the feasible set of $\omega^{[m-1]} = (\omega_{i_1 \ldots i_{m-1}})$, where $i_1 = 1, \ldots, N; \ldots; i_{m-1} = 1, \ldots, N$ satisfying the constraints of the $(m-1)$-dimensional assignment problem:

$$\omega_{i_1 \ldots i_{m-1}} \in \{0, 1\}$$

$$\sum_{i_2=1}^{N} \cdots \sum_{i_{m-1}=1}^{N} \omega_{i_1 \ldots i_{m-1}} = 1, \forall i_1 \in \{1, ..., N\},$$

$$\ldots,$$

$$\sum_{i_1=1}^{N} \cdots \sum_{i_{m-2}=1}^{N} \omega_{i_1 \ldots i_{m-1}} = 1, \forall i_{m-1} \in \{1, ..., N\}.$$

The LR method operates by maximizing the following objective function:

$$G^{[1]}(\mu_M) = \min_{\omega^{[M-1]} \in S^{[M-1]}} \Big\{ \sum_{i_1=1}^{N} \cdots \sum_{i_{M-1}=1}^{N} c^{[M-1]}_{i_1 \ldots i_{M-1}} \omega_{i_1 \ldots i_{M-1}} \Big\} + \sum_{k=1}^{N} \mu_{M,k}, \tag{3.4.1}$$

where $\omega^{[M-1]} = (\omega_{i_1 \ldots i_{M-1}})$,

$$c^{[M-1]}_{i_1 \ldots i_{M-1}} = \min_{1 \le k \le N} \big(c^{[M]}_{i_1 \ldots i_M} - \mu^{(M)}_k \big) \tag{3.4.2}$$

with the convention

$$c^{[M]}_{i_1 \ldots i_M} = c_{i_1 \ldots i_M}. \tag{3.4.3}$$

In the case of $M > 3$, the problem

$$\min_{\omega^{[M-1]}} \Big\{ \sum_{i_1=1}^{N} \cdots \sum_{i_{M-1}=1}^{N} c^{[M-1]}_{i_1 \ldots i_{M-1}} \omega_{i_1 \ldots i_{M-1}} \Big\} \tag{PP(M-1))}$$

is still NP-hard, and hence needs further relaxation.

By repeating the relaxation process, we will end up with a maximazation problem:

$$\max_{\mu_2, \ldots, \mu_M} G^{[M-1]}(\mu_2, \ldots, \mu_M), \tag{3.4.4}$$

where

$$\begin{aligned} & G^{[M-1]}(\mu_2, \ldots, \mu_M) \\ &= \min_{\omega^{[2]} \in S^{[2]}_F} \Big\{ \sum_{i_1=1}^{N} \sum_{i_{M-1}=1}^{N} c^{[2]}_{i_1 i_2} \omega_{i_1 i_2} \Big\} + \sum_{m=2}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m}, \end{aligned} \tag{3.4.5}$$

with $\omega^{[2]} = (\omega_{i_1 i_2})$, and the coefficients $c^{[m]}_{i_1 \dots i_m}$ are recursively defined by

$$c^{[m-1]}_{i_1 \dots i_{m-1}} = \min_{1 \le i_m \le N} \big(c^{[m]}_{i_1 \dots i_m} - \mu_{m,i_m} \big). \tag{3.4.6}$$

for $m = 3, \dots, M$.

The following representation can be derived from (3.4.6) by induction.

Lemma 3.4.1.

$$\begin{aligned} & G^{[M-1]}(\mu_2, \dots, \mu_M) \\ &= \min_{\omega^{[2]} \in S_F^{[2]}} \sum_{i_1=1}^{N} \Big\{ c_{i_1 \dots i_M} - \sum_{m=3}^{M} \mu_{m,i_m} \Big\} \omega_{i_1 i_2} + \sum_{m=3}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m}. \end{aligned} \tag{3.4.7}$$

Let us denote the optimal value of the LR iterations by

$$J^*_{LR} = \max_{\mu_2, \dots, \mu^{(M)}} G^{[M-1]}(\mu_2, \dots, \mu_M). \tag{3.4.8}$$

Moreover, define

$$K^* = \max_{\mu \in F^{[M]}} \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \tag{3.4.9}$$

where the set $F^{[M]}$ is given by

$$F^{[M]} = \{ \mu : \sum_{m=1}^{M} \mu_{m,i_m} \le c_{i_1 \dots i_M}, \text{ for } \forall (i_1, \dots, i_M) \in I_0 \} \tag{3.4.10}$$

Lemma 3.4.2. *For any $\tilde{\mu}_2, \dots, \tilde{\mu}_M$, we have*

$$G^{[M-1]}(\tilde{\mu}_2, \dots, \tilde{\mu}_M) \le K^* \tag{3.4.11}$$

Proof. We know that

$$\begin{aligned} & G^{[M-1]}(\tilde{\mu}_2, \ldots, \tilde{\mu}_M) \\ = & \min_{\omega^{[2]} \in S_F^{[2]}} \sum_{i_1=1}^{N} \Big\{ c_{i_1 \ldots i_M} - \sum_{m=3}^{M} \tilde{\mu}_{m,i_m} \Big\} \omega_{i_1 i_2} + \sum_{m=3}^{M} \sum_{i_m=1}^{N} \tilde{\mu}_{M,i_M}. \end{aligned} \tag{3.4.12}$$

Thus, from 2-D assignment theory,

$$\begin{aligned} & G^{[M-1]}(\tilde{\mu}_2, \ldots, \tilde{\mu}_M) \\ = & \max_{(\mu_{1,i_1},\ \mu_{2,i_2}) \in F^{[2]}} \Big\{ \sum_{i_1=1}^{N} \mu_{1,i_1} + \sum_{i_2=1}^{N} \mu_{2,i_2} \Big\} + \sum_{m=3}^{M} \sum_{i_m=1}^{N} \tilde{\mu}_{m,i_m}, \end{aligned} \tag{3.4.13}$$

where $F^{[2]}$ is the set of $(\mu_{1,i_1}, \mu_{2,i_2})$ satisfying the inequalities

$$\sum_{m=1}^{2} \sum_{i_m=1}^{N} \mu_{m,i_m} \leq c_{i_1 \ldots i_M} - \sum_{m=3}^{M} \tilde{\mu}_{m,i_m} \tag{3.4.14}$$

for any $(i_1,\ i_2) \in \{1, \ldots, N\}^2$. □

It follows that

$$\begin{aligned} & G^{[M-1]}(\tilde{\mu}_2, \ldots, \tilde{\mu}_M) \\ \leq & \max_{\mu \in F^{[M]}} \Big\{ \sum_{i_1=1}^{N} \mu_{1,i_1} + \sum_{i_2=1}^{N} \mu_{2,i_2} + \sum_{m=3}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \Big\} \\ = & K^*, \end{aligned} \tag{3.4.15}$$

since $(\mu_{1,i_1}, \mu_{2,i_2}, \tilde{\mu}_{3,i_3}, \ldots, \tilde{\mu}_{M,i_M}) \in F^{[M]}$, due to inequality (3.4.14). □

Lemma 3.4.3. *Let H^* be the optimal value of the problem (PD). Then, there exists*

$\mu^* \in F^{[M]}$, *such that*

$$\begin{aligned} H^* =& \Big\{ c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \mu^*_{m,i_m} \Big\}^- + \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu^*_{m,i_m} \\ =& \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu^*_{m,i_m} \end{aligned} \tag{3.4.16}$$

Proof. Suppose that $\tilde{\mu}^*$ is an optimal solution to problem (PD). By Lemma 3.3.3, for any $i_1 = 1, \ldots, M$, there is at most one index $(i_1, \ldots, i_M) \in I_0$, with

$$c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \tilde{\mu}^*_{m,i_m} < 0. \tag{3.4.17}$$

Therefore, by rearranging the indices, if necessary, we can assume that

$$c_{k \ldots k} - \sum_{m=1}^{M} \tilde{\mu}^*_{m,k} < 0$$

for $k = 1, ..., N_1$, with $N_1 \leq N$, while $c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \tilde{\mu}^*_{m,i_m} \geq 0$ for all other indices. Define $\mu^* = (\mu^*_{m,i_m})$ by

$$\mu^*_{m,i_m} = \begin{cases} \tilde{\mu}^*_{m,i_m} + c_{k \ldots k} - \sum_{m=1}^{M} \tilde{\mu}^*_{m,k}, & \text{if } m = 1, i_m = 1, ..., N_1, \\ \tilde{\mu}^*_{m,i_m}, & \text{otherwise.} \end{cases} \tag{3.4.18}$$

It can be directly verified that $\mu^* \in F^{[M]}$, and

$$H^* = \Big\{ c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \mu^*_{m,i_m} \Big\}^- + \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu^*_{m,i_m}$$

Therefore, we have established the existence of μ^* claimed by the lemma. □

□

Lemma 3.4.4. *Let H^* be the optimal value of the problem (PD). Then, we have*

$$H^* = K^* \tag{3.4.19}$$

Proof. $\forall \mu \in F^{[M]}$, we have

$$\sum_{m=1}^{M} \mu_{m,i_m} \leq c_{i_1 \ldots i_M} \tag{3.4.20}$$

for any $(i_1, \ldots, i_M) \in \{1, \ldots, N\}^M$.

As a result,

$$\big\{c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \mu_{m,i_m}\big\}^- = 0 \tag{3.4.21}$$

for any $(i_1, \ldots, i_M) \in \{1, \ldots, N\}^M$.

Thus, for any $\mu \in F^{[M]}$,

$$\begin{aligned} &\sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \\ =&\big\{c_{i_1 \ldots i_M} - \sum_{m=1}^{M} \mu_{m,i_m}\big\}^- + \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \\ \leq& H^*. \end{aligned} \tag{3.4.22}$$

It follows that

$$K^* = \max_{\mu \in F^{[M]}} \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \leq H^* \tag{3.4.23}$$

On the other hand, by Lemma 3.4.3, there exists $\mu^* \in F^{[M]}$, such that

$$\begin{aligned} H^* &= \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu^*_{m,i_m} \\ &\leq K^* \end{aligned} \tag{3.4.24}$$

The conclusion of the lemma then follows from (3.4.23) and (3.4.24). □

□

Combining Lemma 3.4.2 and the previous lemma, we get the following

Theorem 3.4.5. *The optimal value H^* of the problem (PD) is an upper bound of J^*_{LR} of the relaxation method, i.e.,*

$$J^*_{LR} \leq H^* \tag{3.4.25}$$

Remark: In fact, by Eq. (3.4.13) and Eq. (3.4.16), we can get

$$J^*_{LR} = H^*. \tag{3.4.26}$$

With respect to the above relationship, we conclude that the LR method and the dual approach are equivalent.

On the other hand, by Lemma 3.4.4, the optimal duality coefficient μ^* can be found by solving the LP problem:

$$\max_{\mu \in F^{[M]}} \sum_{m=1}^{M} \sum_{i_m=1}^{N} \mu_{m,i_m} \tag{3.4.27}$$

where $F^{[M]}$ is the set of μ satisfying the inequalities:

$$\sum_{m=1}^{M} \mu_{m,i_m} \leq c_{i_1 \ldots i_M} \tag{3.4.28}$$

for any $(i_1, \ldots, i_M) \in \{1, \ldots, N\}^M$.

By the duality theory of the linear programming, we have the following theorem.

Theorem 3.4.6. *The optimal solution to the dual problem (PD) can be found by solving the LP problem*

$$\min_x f(x)$$

subject to the constraints:

$$x \geq 0$$

$$\sum_{i_2=1}^{N} \cdots \sum_{i_M=1}^{N} x_{i_1 \ldots i_M} = 1, \forall i_1 \in \{1, ..., N\},$$

$$\sum_{i_1=1}^{N} \cdots \sum_{i_{m-1}=1}^{N} \sum_{i_{m+1}=1}^{N} \cdots \sum_{i_M=1}^{N}, \forall i_m \in \{1, ..., N\},$$

$$\sum_{i_1=1}^{N} \cdots \sum_{i_{M-1}=1}^{N} x_{i_1 \ldots i_M} = 1, \forall i_M \in \{1, ..., N\}.$$

Remark: By the equivalence of the LR method and the dual approach, we observe that LR is equivalent to relaxing the binary constraint condition.

3.5 The solution to the original assignment problem

If the conditions of Theorem 3.3.1 or Theorem 3.3.2 are satisfied, the optimum value of the problem (PP) is equal to the optimal value of the dual problem. Meanwhile, if Theorem 3.3.1 can be applied, the optimal solution $x^* = (x^*_{i_1 \ldots i_M})$ to the original assignment problem is given by

$$x^*_{i_1 \ldots i_M} = \begin{cases} 1, \text{ if } (i_1, \ldots, i_M) = (i_{1,n}, \ldots, i_{M,n}), n = 1, \ldots, N; \\ 0, \text{ otherwise.} \end{cases} \tag{3.5.1}$$

If the conditions of Theorem 3.3.2 are satisfied, the solution is

$$x^*_{i_1 \ldots i_M} = \begin{cases} 1, \text{ if } d^*_{i_1 \ldots i_M} < 0; \\ 0, \text{ if } d^*_{i_1 \ldots i_M} > 0. \end{cases} \tag{3.5.2}$$

On the other hand, if the conditions of the theorems are not satisfied, then we need to search for the optimal (or sometimes sub-optimal) solution. One way is to use the branch-and-bound method [28]. Another way of finding the solution is an $(M-1)$-step procedure shown in Algorithm 1, as done in the LR approach [15, 33, 11, 38],

The complete algorithm for the multidimensional assignment is shown in Algorithm 4.

Remarks: Due to numerical errors, the solution μ^* might not be an optimal solution to (PD). This motivates the above local search operation to mitigate the effect of numerical errors and possible instability in the subgradient iterations.

procedure LRSEARCH(c, μ^*)

Step 1: Solve the relaxed 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} c_{i_1 i_2}^{[2]} \omega_{i_1 i_2} \tag{3.5.3}$$

where $S_F^{[2]}$ is defined in Section 3.4, and

$$c_{i_1 i_2}^{[2]} = \min_{i_3,\ldots,i_M} \big(c_{i_1 \ldots i_M} - \sum_{m=3}^{M} \mu_{m,i_m}^* \big), \tag{3.5.4}$$

Step 2: Fix the assignment $(i_1, i_2(i_1))$ obtained in the previous step, and solve another 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=1}^{N} \sum_{i_3=1}^{N} c_{i_1, i_2(i_1), i_3}^{[3]} \omega_{i_1 i_3} \tag{3.5.5}$$

to obtain the 3-D assignment $(i_1, i_2(i_1), i_3(i_1))$, where

$$c_{i_1, i_2(i_1), i_3}^{[3]} = \min_{i_4,\ldots,i_M} \big(c_{i_1, i_2(i_1), i_3, i_4, \ldots, i_M} - \sum_{m=4}^{M} \mu_{m,i_m}^* \big). \tag{3.5.6}$$

....,

Step $M-1$: Fix the assignment $(i_1, i_2(i_1), \ldots, i_{M-1}(i_1))$ obtained in the previous step, and solve the 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=1}^{N} \sum_{i_3=1}^{N} c_{i_1, i_2(i_1), \ldots i_{M-1}(i_1), i_M} \omega_{i_1 i_M} \tag{3.5.7}$$

to obtain the M-D assignment $(i_1, i_2(i_1), \ldots, i_M(i_1))$.

return the assignment

$$x_{i_1 \ldots i_M}^* = \begin{cases} 1, \text{ if } (i_1, \ldots, i_M) = (i_1, i_2(i_1), \ldots, i_M(i_1)), \text{ for } i_1 = 1, \ldots, N; \\ 0, \text{ otherwise.} \end{cases} \tag{3.5.8}$$

end procedure

Algorithm 1: LR Search algorithm

$\mu^* \leftarrow$ an optimal solution to the problem (PD);
$x^{[0]} \leftarrow$ LRsearch(c,μ^*);
$f^{[0]} \leftarrow f(x^{[0]})$;
$\delta \leftarrow f^{[0]} - H^*$, where $L^* = L(\mu^*)$;
for $k \leftarrow 1, N$ **do** `/* local search */`
  $\mu \leftarrow (\mu_{1,1}, \ldots, \mu_{1,N}, \ldots, \mu_{M,1}, \ldots, \mu_{M,N})$; `/*`

$$\mu_{i,j} = \begin{cases} \mu^*_{i,j} - \delta, & \texttt{if } (i,j) = (M,k) \\ \mu^*_{i,j}, & \texttt{otherwise,} \end{cases} \tag{3.5.9}$$

  `*/`
  $x^{[k]} \leftarrow$ LRsearch(c,μ);
  $f^{[k]} \leftarrow f(x^{[k]})$;
end
return the assignment $x^{[k_0]}$ with $k_0 = \arg\min_{k=0}^{M} f^{[k]}$.

Algorithm 2: The top level function of the multidimensional assignment algorithm

3.6 Simulation results

In this section, we present some simulation results. Our computation environment is a laptop with an Intel i7-6500U CPU running Windows 10. The programming language is MATLAB. The proposed algorithm is tested on four-dimensional assignment problems with known optimal solutions, randomly generated using the method proposed in [18]. The generated problems have dense integer cost coefficients in the range $(0, 20)$. Although the cost coefficients in typical tracking applications are not always dense[33, 11, 38], they can be used to assess the performance of general assignment algorithms. Problems of four different sizes are considered, from relatively small 10x10x10x10 problems to relatively large 40x40x40x40 problems. The choice of the sizes is based on[25], oriented towards tracking applications. Meanwhile, the same set of problems are solved using the LR method as the benchmark to evaluate the performance of the dual method.

The statistics of the randomly generated assignment problems, together with that of the differences between the assignment results, are shown in Table 3.6.1 and Table 3.6.2. Moreover, Fig. 3.6.1 to Fig. 3.6.3 show the distribution of the 20x20x20x20 assignment problems using histograms, together with the distributions of the resulting assignment values.

The subgradient method [42] is used to solve the maximization problems produced by both the dual method and the LR method. Before applying the subgradient method, a normalization is performed on the cost coefficients so that the normalized cost values are in the range $(0, 1)$. The iteration scheme is as follows:

$$x_{n+1} = x_n + \frac{1}{n}\frac{g(x)}{||g(x)||}, \tag{3.6.1}$$

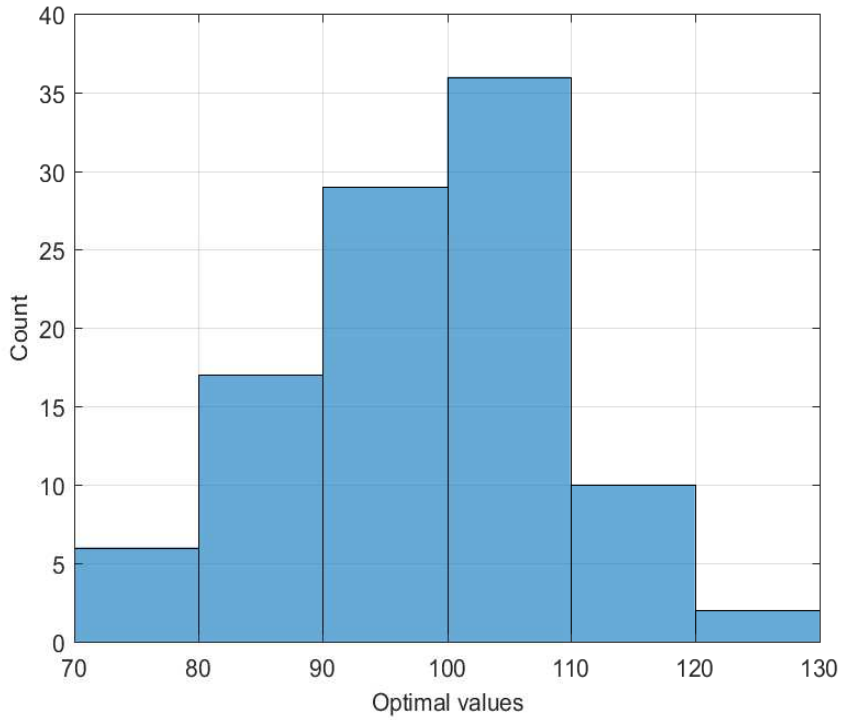

Figure 3.6.1: Distribution of the optimal values of the 100 randomly generated 20x20x20x20 test problems.

Problem size	Optimal value	Difference 1	Difference 2
10x10x10x10	49.1200	2.8470	8.1200
20x20x20x20	98.3200	3.8217	9.7200
30x30x30x30	148.6500	3.8892	9.5500
40x40x40x40	200.0000	5.7897	8.1000

Table 3.6.1 Means of the optimal assignment costs (column 2), the duality gaps, that is, the differences between the optimal values and the dual lower bounds(column 3) and the differences between the actual assignment costs and the optimal values (column 4) for different problem sizes.

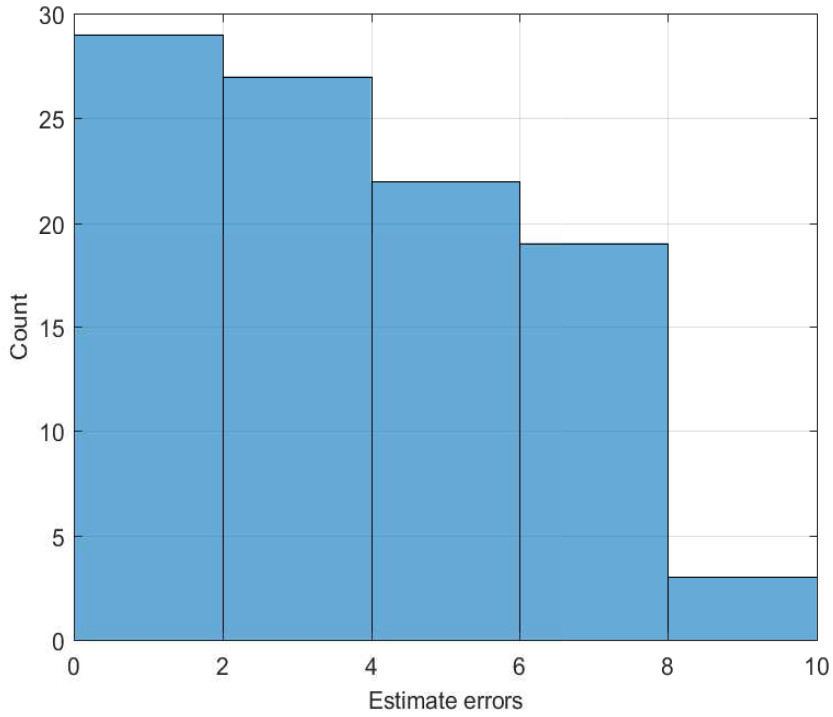

Figure 3.6.2: Distribution of the duality gaps of the 100 randomly generated 20x20x20x20 test problems.

Problem size	Optimal value	Difference 1	Difference 2
10x10x10x10	66.7935	3.8505	9.8844
20x20x20x20	114.3006	5.0702	8.9915
30x30x30x30	171.6079	5.4183	6.8921
40x40x40x40	197.7778	5.4354	7.6556

Table 3.6.2 Variances of the optimal assignment costs (column 2), variances of the duality gaps (column 3) and variances of the differences between actual assignment costs and optimal values (column 4) for different problem sizes.

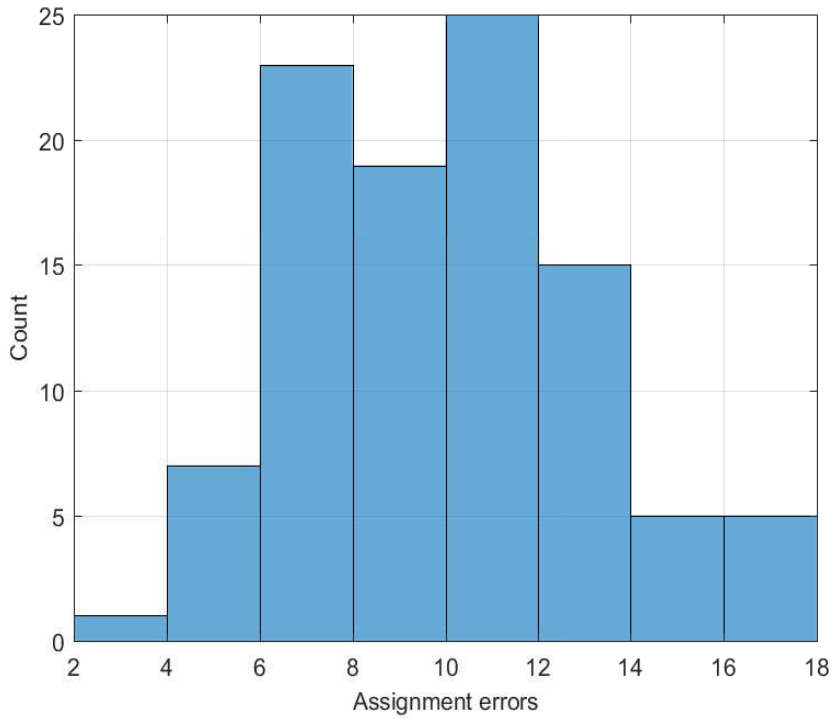

Figure 3.6.3: Distribution of the differences between the assignment results and the optimal values of the 100 randomly generated 20x20x20x20 test problems.

Problem size	Average relative error(%)		CPU time for each problem (s)	
	Dual	LR	Dual	LR
10x10x10x10	9.386396	18.321642	0.03748	0.1081
20x20x20x20	6.06388	10.0953	0.5473	0.8317
30x30x30x30	4.02538	7.30202	3.9443	4.9205
40x40x40x40	2.639375	3.440335	25.6986	28.4049

Table 3.6.3 Relative errors and elapsed CPU times for different problem sizes

where $g(x) \in \partial f(x)$, the subgradient of $f(x)$ at point x. Note that here we want to maximize the objective function $f(x)$. The reason for choosing such a simple method is to ensure that the differences in the simulation results for the two algorithms under tests are indeed caused by the inherent differences between the dual algorithm and the LR algorithm, instead of being produced by the methods used to solve the associated maximization problems.

For the LR method, the stopping criterion adopted in the simulations is

$$\frac{A_n - G_n}{G_n} < 1\%, \tag{3.6.2}$$

with an upper limit of 200 on the number of iterations, where A_n is the total cost corresponding to the assignment result and G_n is the lower bound estimate at the nth step.

For the dual method, the stopping criterion is $n > 100$ and

$$|H_{n+1} - H_n| < 0.0001 H_n, \tag{3.6.3}$$

with an upper limit of 200 on the number of iterations, where H_n is the lower bound estimation at the nth step.

The results are shown in Fig. 1 to Fig. 4, and are summarized in Table 3.6.3.

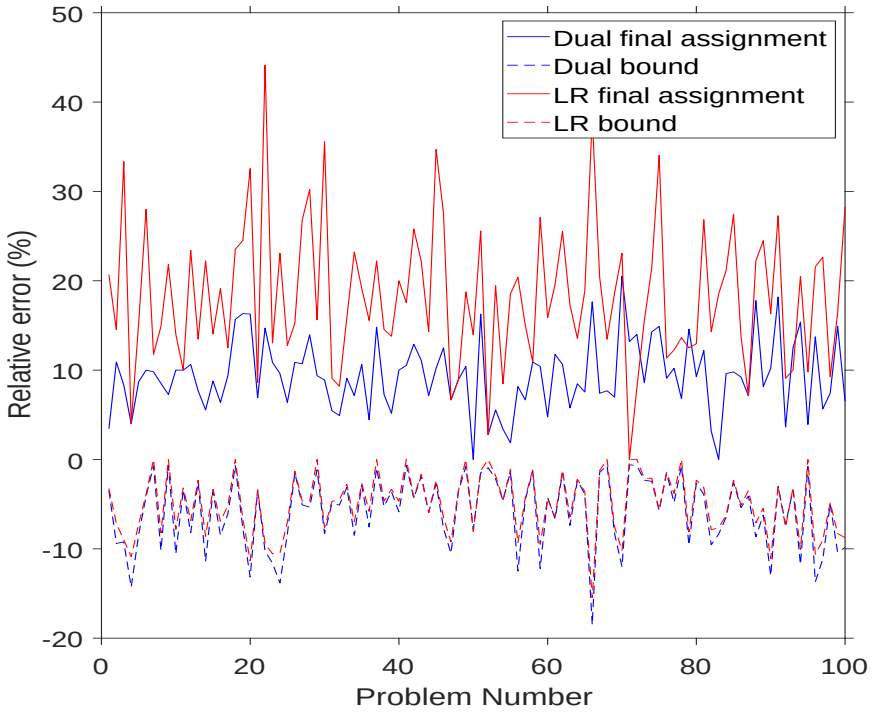

Figure 3.6.4: Relative errors of the assignment results and the lower bound estimates from one hundred randomly generated 10x10x10x10 test problems ($N = 10$).

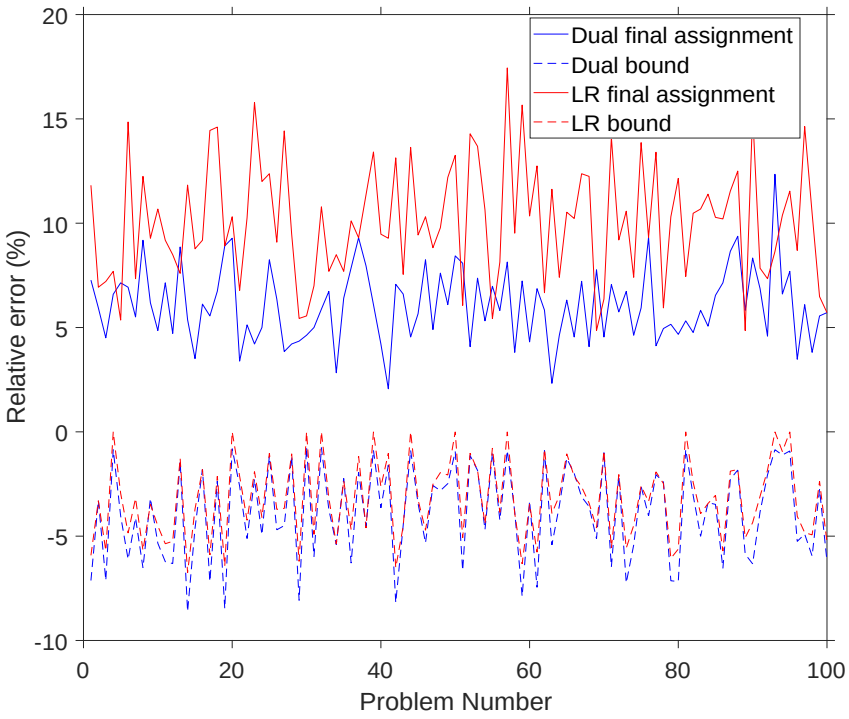

Figure 3.6.5: Relative errors of the assignment results and the lower bound estimates from one hundred randomly generated 20x20x20x20 test problems ($N = 20$).

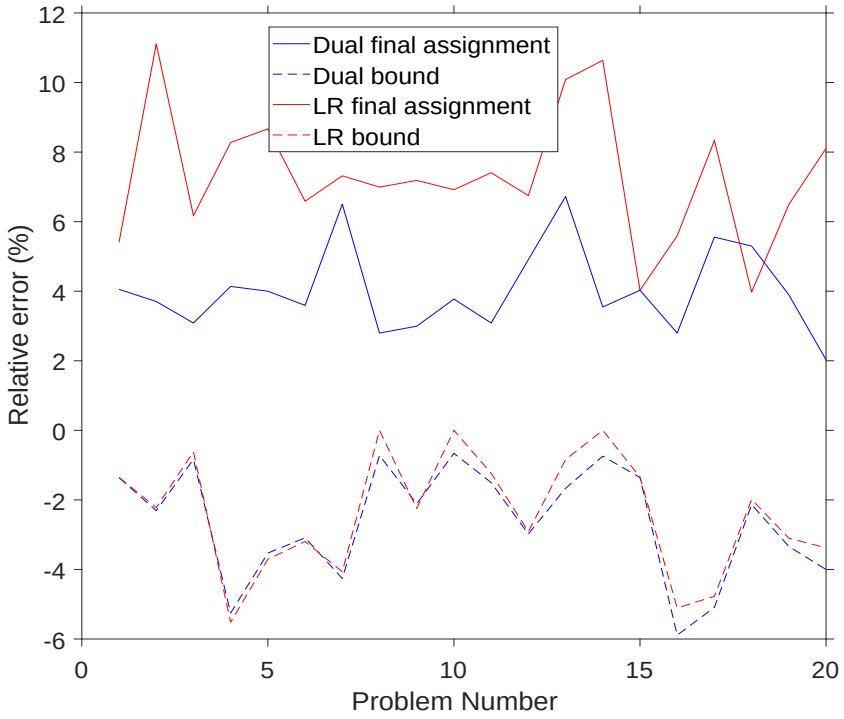

Figure 3.6.6: Relative errors of the assignment results and the lower bound estimates from twenty randomly generated 30x30x30x30 test problems ($N = 30$).

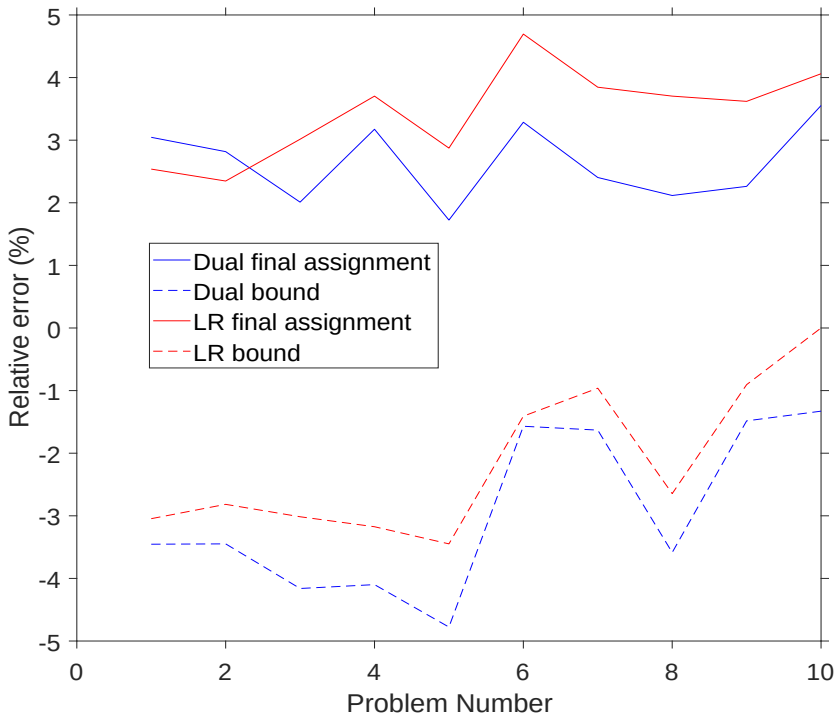

Figure 3.6.7: Relative errors of the assignment results and the lower bound estimates from ten randomly generated 40x40x40x40 test problems ($N = 40$).

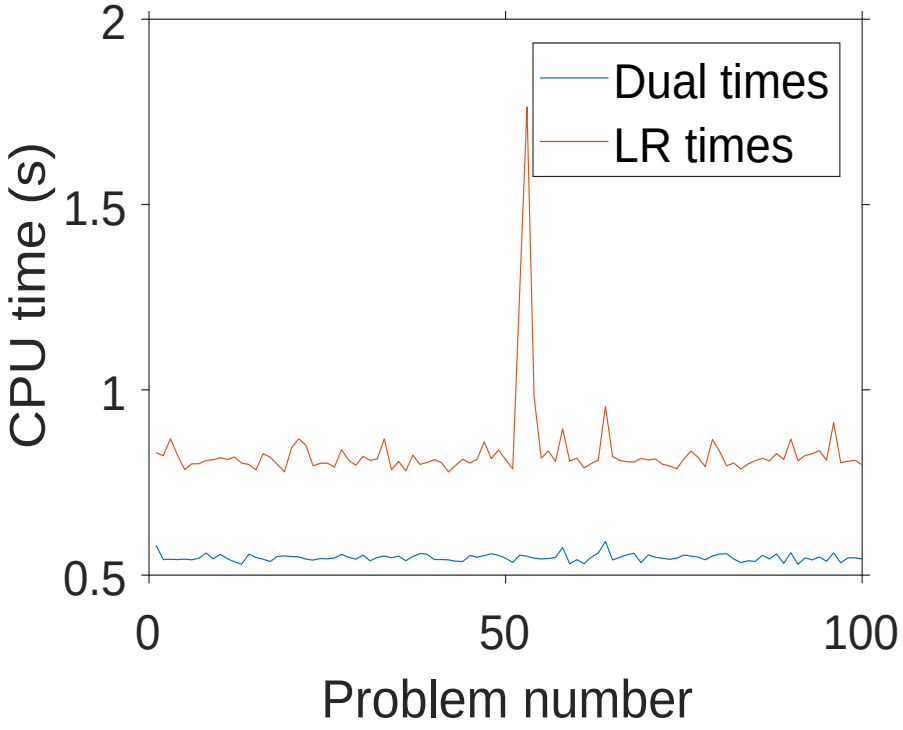

Figure 3.6.8: CPU times for the 100 randomly generated 20x20x20x20 test problems.

The results demonstrate that the dual approach outperforms the LR algorithm both in terms of the elapsed CPU times and the relative accuracy of the final assignments.

The CPU times for the 20x20x20x20 simulation are shown in Fig. 3.6.8, with a mean value of 0.5473s and a standard deviation of 0.0098s. Also shown in the figure are the CPU times of the LR method, which have a standard variation of 0.1094s. The curves of CPU times for other problem sizes have similar characteristics.

On the other hand, the statistics in Table 3.6.1 and Table 3.6.2 and in Fig. 3.6.1 to Fig. 3.6.3 show that after applying the dual algorithm, the variances of original distributions have been reduced significantly. From a statistical perspective, the dual algorithm can be interpreted as a filter that reduces the "noise" in the original data.

3.7 Alternating direction iteration scheme

The time complexity of solving the dual problem (PD) using the subgradient algorithms is proportional to the number of subgradient operations, and thus is $O(N^M)$.

For the LR method, the most time consuming operation is calculating the coefficients

$$c_{i_1 \ldots i_{m-1}}^{[m-1]} = \min_{1 \le i_m \le N} \left(c_{i_1 \ldots i_m}^{[m]} - \mu_{m,i_m} \right). \tag{3.7.1}$$

This operation also has an N^M order time complexity. Since the calculation of the subgradient involves a similar comparison operation, we thus conclude that the dual method and the LR method have the same order of time complexity.

However, in contrast to the recursive calls to the 2-D auction algorithm, which has an $O(N^2)$ time complexity [7], the dual approach can directly get the optimal dual variables by employing a simple subgradient algorithm, and thus is found to be more efficient in our simulations.

On the other hand, as pointed out in Section 3.4, the optimal dual coefficients μ^* can be obtained by solving an Linear programming (LP) problem using any efficient LP algorithm. An alternate direction iteration method, which is essentially a dimension-reduction technique, can efficiently accelerate the solution process of the problem (PD). We illustrate the algorithm for the case of $M = 4$, while pointing out that the idea can be adapted to any value of $M \ge 3$.

The algorithm is based on the following observation:

$$
\begin{aligned}
&\max_{\sum_{m=1}^{4}\mu_{m,i_m}\le c_{i_1\dots i_4},\forall(i_1,\dots,i_4)\in I_0} \sum_{m=1}^{4}\sum_{i_m=1}^{N}\mu_{m,i_m} \\
=&\max_{\mu_3,\mu_4}\Big\{\max_{\mu_{1,i_1}+\mu_{2,i_2}\le c_{i_1\dots i_4}-\mu_{3,i_3}-\mu_{4,i_4},\forall(i_1,\dots,i_4)\in I_0} \sum_{m=1}^{2}\sum_{i_m=1}^{N}\mu_{m,i_m} \\
&+ \sum_{m=3}^{4}\sum_{i_m=1}^{N}\mu_{m,i_m}\Big\} \qquad (3.7.2)\\
=&\max_{\mu_3,\mu_4}\Big\{\max_{\mu_{1,i_1}+\mu_{2,i_2}\le c^{[2]}(i_1,i_2),\forall(i_1,i_2)} \sum_{m=1}^{2}\sum_{i_m=1}^{N}\mu_{m,i_m} \\
&+ \sum_{m=3}^{4}\sum_{i_m=1}^{N}\mu_{m,i_m}\Big\}
\end{aligned}
$$

where $c^{[2]}(i_1,i_2)=\min_{i_3,i_4}\{c_{i_1\dots i_4}-\mu_{3,i_3}-\mu_{4,i_4}\}$, as defined before.

Thus, the four-dimensional maximization operation can be carried out by performing a two-stage iterative maximization, with the inner maximization being a two-dimensional assignment problem. Consequently, we get an alternating direction iteration scheme as shown in Algorithm 3.

The problems in Section 3.6 are recalculated using Algorithm 3. The results are summarized in Table 3.7.1, where we compare the average errors and the elapsed CPU times for the original four-dimensional subgradient method and the two-dimensional alternating direction iteration method described in Algorithm 3. It can be seen that the CPU times have been significantly reduced by the dimension-reduction technique while maintaining the same accuracy, especially for large-size problems. For small-size problems, the original four-dimensional (4D) subgradient method is more efficient than the two-dimensional (2D) iteration method.

$n_{it} \leftarrow$ number of iterations;

$I_2 \leftarrow \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$;
for $k = 1 : n_{it}$ **do**
// External iteration
for (p,q) *in* I_2 **do**
// For every 2D combination
$\{r,s\} \leftarrow I_2 \backslash \{p,q\}$ // Set difference
$c^{[2]}_{i_p,i_q} = \min_{i_r,i_s}\{c_{i_p i_q i_r i_s} - \mu_{r,i_r} - \mu_{s,i_s}\}$,
▷ Solve the following 2D assignment problem
$(v^*, w^*) \leftarrow \arg\max_{v,w} \sum_{i,j}(c^{[2]}_{ij} - v_i - w_j)^- + \sum_i v_i + \sum_j w_j$
▷ $v = (v_1, \ldots, v_N), w = (w_1, \ldots, w_N)$
$\mu_p \leftarrow v^*$
$\mu_q \leftarrow w^*$
end
end

return optimal solution $(\mu_1, \mu_2, \mu_3, \mu_4)$ to problem (PD).

Algorithm 3: Alternating direction iteration algorithm

Problem size	Average relative error(%)		CPU time for each problem (s)	
	4D iteration	2D iteration	4D iteration	2D iteration
10x10x10x10	9.386396	10.162819	0.03748	0.04864
20x20x20x20	6.06388	5.931272	0.58748	0.27613
30x30x30x30	4.02538	3.987029	3.9443	1.59149
40x40x40x40	2.639375	2.605811	25.6986	9.05326

Table 3.7.1 Relative errors and elapsed CPU times for two different iteration methods. 4D iteration corresponds to the subgradient method described in Section 3.6, while 2D iteration corresponds to Algorithm 3, with iteration number n_{it}=10. The subgradient method is used to solve each of the 2D assignment sub-problems in 200 iterations

3.8 Conclusions

In this chapter, a dual approach was proposed for the MAP problem. Moreover, two strong duality results were proved. The dual approach was shown to be closely related to the LR method, providing a framework to conduct a theoretical investigation of the LR algorithm. It was also pointed out that the LR relaxation is equivalent to relaxing the binary constraint condition, and as a result, the dual coefficient μ can be obtained by solving an LP problem.

On the other hand, it was demonstrated that a simple but efficient numerical method can be developed based on the dual formulation, with the same or even better accuracy than that of the LR method. The proposed local search strategy was shown to be effective in the simulations.

Chapter 4

Multi-frame tracking based on multi-dimensional assignment algorithm

4.1 Introduction

The common way of tracking multiple objects is to estimate the tracks of the objects using a single scan or frame of measurements at every time step. But this is not effective in tracking maneuvering targets and multiple targets moving along closely spaced near-parallel tracks. In such scenarios, instead of relying on a single frame of measurements, it is more desirable to consider multiple frames of measurements at every estimation step by taking advantage of the time correlation among the successive data frames. This has attracted a lot of attention in the tracking research community. See [24, 8] for more about multi-frame related developments.

For multiple frame tracking to work, we need to address the problem of associating the measurements in two or more data frames to the targets being tracked. This is a situation where Multidimensional assignment problem (MAP) can play an important role: the multi-frame data association problem can be formulated into an MAP problem, see [4, 46, 24, 36]. The challenge now is to find effective methods to solve the MAP problem.

In the case of single frame tracking, the data association operation can be modeled as a 2-D data association problem, for which a number of efficient algorithms have been proposed in the literature. See [7] and the references therein. Unlike 2-D assignment problems, however, MAP is an NP-hard problem [16]. As a consequence, it is impractical to get exact solutions to large scale MAPs for applications with real time requirements as in some surveillance systems. A number of approximate MAP assignment algorithms have been proposed [33, 37, 3, 11]. These algorithms are much more efficient in terms of the CPU time than the exact methods such as the branch and bound algorithm [40]. Among these, the Lagrangian relaxation method has been applied to multi-object tracking with good association accuracy [33, 37, 11]. Therefore, multi-frame assignment is a promising method that merits further research to meet the requirements of practical tracking applications.

We recently developed a novel multidimensional assignment algorithm based on the idea of dual convexification [31, 30].The newly proposed algorithm is shown to be equivalent to the Lagrangian relaxation method in the sense that they both provide the same lower bound estimation for the optimal total assignment cost. However, the dual algorithm is much more elegant in the formulation and easier to implement than

the Lagrangian method. This is a significant advantage in practical engineering applications. Our extensive simulation results in [31, 30] showed that the dual algorithm is more efficient than the Lagrangian relaxation method in solving randomly generated MAP problems. Now we use this new algorithm to solve assignment problems arising from a series of multi-frame tracking problems with different complexities. The results demonstrate that the dual method outperforms the Lagrangian method in this context, while maintaining the same level of high accuracy.

4.2 Problem formulation

We consider the problem of tracking multiple objects in a plane. The state space model of each of the objects is

$$\begin{aligned}\begin{pmatrix} x(k+1) \\ y(k+1) \\ u_x(k+1) \\ u_y(k+1) \end{pmatrix} =& F \begin{pmatrix} x(k) \\ y(k) \\ u_x(k) \\ u_y(k) \end{pmatrix} \\ &+ \begin{pmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{pmatrix} \begin{pmatrix} v_x(k) \\ v_y(k) \end{pmatrix}\end{aligned} \tag{4.2.1}$$

where

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2.2}$$

$(x(k), y(k)$ and $(u_x(k), u_y(k)$ are the position and velocity vectors of the target at time k, respectively, while $(v_x(k), v_y(k)$ is the acceleration vector caused by Gaussian white noise at time k. T is the time step.

The measurement model is given by

$$r(k) = \sqrt{x^2(k) + y^2(k)} + w_r(k) \tag{4.2.3}$$

$$\theta(k) = \arctan(\frac{y(k)}{x(k)}) + w_\theta(k) \tag{4.2.4}$$

where $(r(k), \theta(k))$ is the polar coordinate of the target at time k, while $w_r(k)$ and $w_\theta(k)$ are the Gaussian white noise components at time k in the range and bearing measurements, respectively.

4.3 2-D assignment formulation for single frame association

Let V be the area of the field of view, which is divided into small cells of area Δv. The field of view is divided into small cells, so that we can more conveniently refer to the probability of a measurement being in a particular cell, instead of being forced to use the probability density in the following formulations. By the routine reasoning, see

for example [4], one can verify that the probability of associating the measurement $z_j(k)$ to target i at time k is

$$\Lambda_{i,j}(k) = \begin{cases} P_F \frac{1}{V} \Delta v & \text{if } i = 0,\ j \neq 0 \\ (1 - P_D) \frac{1}{V} \Delta v & \text{if } i \neq 0,\ j = 0 \\ P_D \mathcal{N}[z_j(k); \hat{z}_i(k|k-1), S_i(k)] \Delta v & \text{otherwise.} \end{cases} \tag{4.3.1}$$

where $\hat{z}_i(k|k-1)$ is the predicted measurement for target i at time k, and $S_i(k)$ is the covariance of the measurement's distribution for target i at time k, V is the area of each cell. Moreover, as usual, the dummy target 0 represents false alarms and new targets, and the dummy measurement 0 represents a missing target.

Remark: The value of $\Lambda_{0,0}(k)$ has no effect on the data association, and can be set to a value which is convenient in the calculations.

Now, let us denote the data association which assign measurement j to target i by the symbol $i \leftrightarrow j$, then the best data associations $i \leftrightarrow j_i, i = 1, \ldots, N_1$ are those which maximize the likelihood product

$$\Lambda_{1,j_1}(k) \Lambda_{2,j_2}(k) \ldots \Lambda_{N_1,j_{N_1}}(k), \tag{4.3.2}$$

where N_1 is the number of confirmed targets, assuming independence of the likelihoods. However, instead of workling on the above likelihoods directly, it turns out to be more convenient to consider the following negative logarithmic likelihood ratio,

see [4],

$$c_{i,j} = -\ln \frac{\Lambda_{i,j}}{\Lambda_{0,j}} \tag{4.3.3}$$

for $(i,j) \in I_2$, with I_2 being an index set given by

$$\begin{aligned} I_2 &= \{0,\ldots,N_1\} \times \{0,\ldots,N_2\} \\ &= \{(i,j) \mid i \in \{0,\ldots,N_1\},\ldots,j \in \{0,\ldots,N_2\}\}. \end{aligned} \tag{4.3.4}$$

.

If $i \neq 0$,

$$\begin{aligned} c_{i,j} = &\ln \frac{P_F(2\pi|S_i(k)|)^{\frac{n}{2}}}{P_D V} \\ &+ \frac{1}{2}(z_j(k) - \hat{z}_i(k|k-1))^T S_i(k)^{-1}(z_j(k) - \hat{z}_i(k|k-1))) \end{aligned} \tag{4.3.5}$$

where n is the dimension of the observation vector. In the simpler case of $i = 0$, $c_{i,j}$ can be obtained from Eq. (4.3.2) correspondingly.

Moreover, define the linear objective function

$$f(\rho) = \sum_{(i,j)\in I_2} c_{i,j}\rho_{i,j}, \tag{4.3.6}$$

where $\rho = (\rho_{i,j})$. It can be seen that the original data association problem can be represented as a 2-dimensional assignment problem:

$$\min_{\rho} f(\rho), \tag{PP}$$

subject to the constraints

$$\rho_{i,j} \in \{0,1\} \tag{4.3.7}$$

$$\sum_{j=0}^{N_2} \rho_{i,j} = 1, \forall i \in \{1,...,N_1\}, \tag{4.3.8}$$

$$\sum_{i=0}^{N_1} \rho_{i,j} = 1, \forall j \in \{1,...,N_2\}. \tag{4.3.9}$$

4.4 3-D assignment formulation for double frame association

Next, we consider the two frame data association problem. For a real target i, namely $i \neq 0$, the probability of associating the measurements $z_j(k)$ and $z_m(k+1)$ to target i at time k is

$$\Lambda_{i,j,m}(k) = \Lambda_{i,j}(k) P_{m|i,j}(k) \tag{4.4.1}$$

where $P_{m|i,j}(k)$ is the probability of associating measurement $z_m(k+1)$ to target i given that the measurement $z_j(k)$ is associated to target i at time k.

As in [24], if $j \neq 0$ and $k \neq 0$, namely for real measurements, $P_{m|i,j}(k)$ can be calculated as follows.

$$P_{m|i,j}(k) = P_D \mathcal{N}[z_m(k+1); \hat{z}_i(k+1|k), S_i(k+1)]\Delta v, \tag{4.4.2}$$

where $\hat{z}_i(k+1|k)$ and $S_i(k+1)$ are calculated by applying Kalman filter on the state of target i at time $k-1$ and taking $z_j(k)$ as the measurement from target i at time k.

The case of $j = 0$ and $k \neq 0$ can be calculated similarly. The only difference is that here we do not update the state $\hat{x}(k-1|k)$ with an actual measurement during the course of applying the Kalman filter to calculate $\hat{z}_i(k+1|k)$.

For $k = 0$, we simply have $P_{m|i,j}(k) = P_F \frac{1}{V} \Delta v$.

Similar to what we have done in the case of single frame associations, we consider the negative logarithmic likelihood ratio, see [4],

$$c_{i,j,m} = -\ln \frac{\Lambda_{i,j,m}}{\Lambda_{0,j,m}} \tag{4.4.3}$$

for $(i, j, m) \in I_3$, with I_3 being an index set given by

$$I_3 = \{0, \ldots, N_1\} \times \{0, \ldots, N_2\} \times \{0, \ldots, N_3\} \tag{4.4.4}$$

.

Moreover, define the linear objective function

$$f(\rho) = \sum_{(i,j,m) \in I_3} c_{i,j,m} \rho_{i,j,m}, \tag{4.4.5}$$

where $\rho = (\rho_{i,j,m})$. It can be seen that the original double frame data association problem can be cast into a 3-dimensional assignment problem:

$$\min_{\rho} f(\rho), \tag{PP}$$

subject to the constraints

$$\rho_{i,j,m} \in \{0,1\} \tag{4.4.6}$$

$$\sum_{j=0}^{N_2}\sum_{m=0}^{N_3} \rho_{i,j,m} = 1, \forall i \in \{1,...,N_1\}, \tag{4.4.7}$$

$$\sum_{i=0}^{N_1}\sum_{m=0}^{N_3} \rho_{i,j,m} = 1, \forall j \in \{1,...,N_2\}. \tag{4.4.8}$$

$$\sum_{i=0}^{N_1}\sum_{j=0}^{N_2} \rho_{i,j,m} = 1, \forall m \in \{1,...,N_3\}. \tag{4.4.9}$$

4.5 Assignment problem for general multi-frame data association

In general, let $M \geq 3$ be an integer, we have similar formulations of $(M-1)$-frame data association problem as M-dimensional assignment problems. The cost coefficients $\{c_{i_1\ldots i_M}\}, (i_1,\ldots,i_M) \in I_0$ are defined as

$$c_{i_1 i_2\ldots i_M} = -\ln \frac{\Lambda_{i_1 i_2\ldots i_M}}{\Lambda_{0 i_2\ldots i_M}}, \tag{4.5.1}$$

and correspondingly, we have a linear objective function

$$f(\rho) = \sum_{(i_1,\ldots,i_M)\in I_0} c_{i_1\ldots i_M}\rho_{i_1\ldots 1_M}, \tag{4.5.2}$$

where $\rho = (\rho_{i_1\ldots i_M})$, and I_0 is an index set given by

$$I_0 = \{0,\ldots,N_1\} \times \{0,\ldots,N_2\} \times \cdots \times \{0,\ldots,N_M\}, \tag{4.5.3}$$

.

The M-dimensional assignment problem corresponding to $M-1$-frame data association problems is

$$\min_{\rho} f(\rho), \tag{PP}$$

subject to the constraints

$$\rho_{i_1 \ldots i_M} \in \{0,1\} \tag{4.5.4}$$

$$\sum_{i_2=0}^{N_2} \cdots \sum_{i_M=0}^{N_M} \rho_{i_1 \ldots i_M} = 1, \forall i_1 \in \{1, ..., N_1\}, \tag{4.5.5}$$

$$\sum_{i_1=0}^{N_1} \cdots \sum_{i_{m-1}=0}^{N_{m-1}} \sum_{i_{m+1}=0}^{N_{m+1}} \cdots \sum_{i_M=0}^{N} \rho_{i_1 \ldots i_M} = 1, \forall i_m \in \{1, ..., N_m\}, \tag{4.5.6}$$

$$\sum_{i_1=0}^{N_1} \cdots \sum_{i_{M-1}=0}^{N_{M-1}} \rho_{i_1 \ldots i_M} = 1, \forall i_M \in \{1, ..., N_M\}. \tag{4.5.7}$$

Similarly, $\rho = (\rho_{i_1 \ldots i_M})$, with $\rho_{i_1 \ldots i_M} = 1$, if measurement z_{m-1,i_m} is associated to track i_1 for $m = 2, \ldots, M$, otherwise, $\rho_{i_1 \ldots i_M} = 0$.

As in the previous chapter, the following vector notations will be adopted in this chapter:

$$\rho = (\rho_{00\ldots0}, \rho_{10\ldots0}, ..., \rho_{N_1 \ldots 0}, ..., \rho_{N_1 \ldots N_M}),$$

$$\mu_1 = (\mu_{1,0}, \mu_{1,1}, ..., \mu_{1,N_1}),$$

$$\ldots, \mu = (\mu_1, \ldots, \mu_M).$$

Defining

$$L(\mu) = \sum_{(i_1,\ldots,i_M) \in I_0} \left(d_{i_1 \ldots i_M}\right)^{-} + \sum_{m=1}^{M} \sum_{i_m=0}^{N_m} \mu_{m,i_m}. \tag{4.5.8}$$

where

$$d_{i_1 \ldots i_M} = c_{i_1 \ldots i_M} - \mu_{1,i_1} - \mu_{2,i_2} - \cdots - \mu_{M,i_M}. \tag{4.5.9}$$

We will consider the following dual problem

$$\max_{\mu} L(\mu), \tag{PD}$$

under the condition that $\mu_{m,0} = 0$ for $m = 1, \ldots, M$.

With the above assignment problem, we have the following theorem [30].

Theorem 4.5.1. *The optimal solution to the dual problem (PD) can be found by solving the LP problem*

$$\min_{\rho} f(\rho)$$

subject to the constraints:

$$\rho \geq 0$$

$$\sum_{i_2=0}^{N_2} \cdots \sum_{i_M=0}^{N_M} \rho_{i_1 \ldots i_M} = 1, \forall i_1 \in \{1, ..., N_1\},$$

$$\sum_{i_1=0}^{N_1} \cdots \sum_{i_{m-1}=0}^{N_{m-1}} \sum_{i_{m+1}=0}^{N_{m+1}} \cdots \sum_{i_M=0}^{N_M}, \forall i_m \in \{1, ..., N_m\},$$

$$\sum_{i_1=0}^{N_1} \cdots \sum_{i_{M-1}=0}^{N_{M-1}} \rho_{i_1 \ldots i_M} = 1, \forall i_M \in \{1, ..., N_M\}.$$

Furthermore, it has been shown that the dual formulation is equivalent to the Largrangian relaxation formulation in terms of their estimations of the lower bound on the optimal assignment value. More specifically, we have the following theorem [30].

Theorem 4.5.2. *Let L^* be the optimal valuethe of problem (PD) , and J^*_{LR} the optimal value of the objective function of the Lagragian relaxation method, then*

$$J^*_{LR} = L^*, \tag{4.5.10}$$

4.6 Algorithm for the assignment problem

The method to find the solution to the M-dimensional assignment problem is an $(M-1)$-step procedure shown in Algorithm 4.

$\mu^* \leftarrow$ an optimal solution to the problem (PD);
$\rho^{[0]} \leftarrow$ LRsearch(c,μ^*);
$f^{[0]} \leftarrow f(\rho^{[0]})$;
$\delta \leftarrow f^{[0]} - H^*$, where $L^* = L(\mu^*)$;
for $k \leftarrow 1, N$ **do** `/* local search */`
  $\mu \leftarrow (\mu_{1,1}, \ldots, \mu_{1,N}, \ldots, \mu_{M,1}, \ldots, \mu_{M,N})$; `/*`

$$\mu_{i,j} = \begin{cases} \mu^*_{i,j} - \delta, & \texttt{if } (i,j) = (M,k) \\ \mu^*_{i,j}, & \texttt{otherwise,} \end{cases} \tag{4.6.1}$$

  `*/`
  $\rho^{[k]} \leftarrow$ LRsearch(c,μ);
  $f^{[k]} \leftarrow f(\rho^{[k]})$;
end
return the assignment $x^{[k_0]}$ with $k_0 = \arg\min_{k=0}^{M} f^{[k]}$.

Algorithm 4: The top level function of the multidimensional assignment algorithm

The reason for the local search operation in Algorithm 4 is that, due to numerical errors, the solution μ^* might not be an optimal solution to (PD). In order to mitigate the effect of numerical errors and possible instability in the subgradient iterations, we perform the above operations. The LRsearch step is given in Algorithm 5, where

$S_F^{[2]}$ represents the constraints in 2-D assignment problems.

4.6.1 Subgradient algorithm implementation

The subgradient method [42] is used to solve the maximization problem (PD). The iteration scheme is given by

$$\mu(n+1) = \mu(n) + \Delta t(n) g(\mu(n)), \tag{4.6.8}$$

where $\Delta t(n)$ is the step size at the nth iteration, and $g(\mu) \in \partial L(\mu)$ is a subgradient of $L(\mu)$ at point μ. Note that here we want to maximize the objective function $H(\mu)$.

In typical tracking problems [25], the cost coefficients are sparse arrays, meaning that a number of cost coefficients are large enough and will not make any contribution in the subgradient calculation. Therefore, these large cost coefficients can be regarded as infinity. This motivate us to use sparse array data structures to represent the costs for both memory and CPU time efficiency, and correspondingly formulate the subgradient algorithm based on sparse array representations. The sparse array can be implemented as a matrix C,

$$C = \begin{bmatrix} 1 & 3 & 2 & c_{132} \\ \cdots & & & \\ i & j & k & c_{ijk} \\ \cdots & & & \end{bmatrix} \tag{4.6.9}$$

where each row represents a cost item: the index (i, j, k) of the cost and its value c_{ijk}.

The pseudo code for 3-D assignment problems is shown in Algorithm 6. In the list of pseudo code, we use (u, v, w) to represent the Lagrangian multipliers (μ_1, μ_2, μ_3)

Function `LRsearch`(c, μ^*)**:**

Step 1: Solve the relaxed 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=0}^{N_1} \sum_{i_2=0}^{N_2} c_{i_1 i_2}^{[2]} \omega_{i_1 i_2} \tag{4.6.2}$$

where

$$c_{i_1 i_2}^{[2]} = \min_{i_3,\ldots,i_M} \big(c_{i_1 \ldots i_M} - \sum_{m=3}^{M} \mu_{m,i_m}^* \big), \tag{4.6.3}$$

Step 2: Fix the assignment $(i_1, i_2(i_1))$ obtained in the previous step, and solve another 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=0}^{N_1} \sum_{i_3=0}^{N_3} c_{i_1, i_2(i_1), i_3}^{[3]} \omega_{i_1 i_3} \tag{4.6.4}$$

to obtain the 3-D assignment $(i_1, i_2(i_1), i_3(i_1))$, where

$$c_{i_1, i_2(i_1), i_3}^{[3]} = \min_{i_4,\ldots,i_M} \big(c_{i_1, i_2(i_1), i_3, i_4, \ldots, i_M} - \sum_{m=4}^{M} \mu_{m,i_m}^* \big). \tag{4.6.5}$$

....,

Step $M-1$: Fix the assignment $(i_1, i_2(i_1), ..., i_{M-1}(i_1))$ obtained in the previous step, and solve another 2-D assignment problem:

$$\min_{\omega \in S_F^{[2]}} \sum_{i_1=0}^{N_1} \sum_{i_3=0}^{N_3} c_{i_1, i_2(i_1), \ldots, i_{M-1}(i_1), i_M}^{[M]} \omega_{i_1 i_M} \tag{4.6.6}$$

to obtain the M-D assignment $(i_1, i_2(i_1), i_3(i_1), ..., i_M(i_1))$, where

$$c_{i_1, i_2(i_1), \ldots, i_{M-1}(i_1), i_M}^{[M]} = \min_{i_4,\ldots,i_M} \big(c_{i_1, i_2(i_1), \ldots, i_{M-1}(i_1), i_M} - \mu_{M,i_M}^* \big). \tag{4.6.7}$$

Step M:
// $i_1 = 1, \ldots, N$
if $(i_1, \ldots, i_M) = (i_1, i_2(i_1), \ldots, i_M(i_1))$ **then**
  $x_{i_1 \ldots i_M}^* = 1$;
end
else
  $x_{i_1 \ldots i_M}^* = 0$;
end
return x^*

Algorithm 5: LR Search algorithm

for clarity.

After getting a solution (u, v, w), we can form another sparse array

$$D = \begin{bmatrix} 1 & 3 & 2 & c_{132} - u_1 - v_3 - w_2 \\ \cdots & & & \\ i & j & k & c_{ijk} - u_i - v_j - w_k \\ \cdots & & & \end{bmatrix} \tag{4.6.10}$$

where each row represents a reduced cost item: the index of the item (i, j, k) and the value $c_{ijk} - u_i - v_j - w_k$.

4.6.2 Step size calculation

One of the key operations in the implementation of the subgradient algorithm is to determine the step size Δt_n. In our simulation, it has been shown that the following method is efficient [10].

$$\Delta t(n) = \frac{\hat{L} - L(n)}{||s(n)||^2}, \tag{4.6.11}$$

where $L(n)$ is the value of the objective function L in the nth iteration step, $\hat{L}$ is the estimate of the optimal value of L, and $s(n)$ is the modified subgradient at the nth step given by the recursive equation

$$s(n) = g(n) + \beta(n) * s(n-1), \tag{4.6.12}$$

Data: The costs in a sparse array C.
$N_I \leftarrow$ number of iterations;
// Initialization
$u(0) \leftarrow 0$; // $u = (u_1, \ldots, u_N)$
$v(0) \leftarrow 0$; // $v = (v_1, \ldots, v_N)$
$w(0) \leftarrow 0$; // $w = (w_1, \ldots, w_N)$
n=1;
while $n \leq N_I$ **do**
  /* The subgradient */
  $\Delta u(n) \leftarrow 1$; $\Delta v(n) \leftarrow 1$; $\Delta w(n) \leftarrow 1$;
  for $\forall (i, j, k, c_{ijk}) \in C$ **do**
    if $c_{ijk} - u_i - v_j - w_k < 0$ **then**
      $\Delta u(n) \leftarrow \Delta u(n-1) - 1$; $\Delta v(n) \leftarrow \Delta v(n-1) - 1$;
      $\Delta w(n) \leftarrow \Delta w(n-1) - 1$;
    end
    /* Solve the following 2D assignment problem */
    Determine the step size $\Delta t(n)$;
    $u(n) \leftarrow u(n-1) + \Delta t(n) * \Delta u(n)$;
    $v(n) \leftarrow v(n-1) + \Delta t(n) * \Delta v(n)$;
    $w(n) \leftarrow w(n-1) + \Delta t(n) * \Delta w(n)$;
    $n = n + 1$;
  end
end
return optimal solution $(u(n), v(n), w(n))$;

Algorithm 6: Subgradient algorithm based on sparse array representation of the costs

where $g(n)$ is a subgradient of $L(\mu)$ at the nth step, and the coefficient $\beta(n)$ is given by

$$\beta(n) = \begin{cases} 0; \text{ if}(s(n-1), g(n)) \geq 0 \\ -\gamma(n)\frac{(s(n-1),g(n))}{||s(n-1)||^2}; \text{ otherwise;} \end{cases} \tag{4.6.13}$$

where $(s(n-1), g(n))$ is the inner product of the two vectors, and $0 \leq \gamma(n) \leq 2$.

The estimate $\hat{L}$ of the optimal value of L, can be obtained by solving 2-D assignment problems. Let us illustrate it with a two frame tracking problem. Suppose we have N_1 tracks at time k, together with two frames of measurement, $z_1(k+1), \ldots, z_{N_2}(k+1)$ for time $k+1$, and $z_1(k+2), \ldots, z_{N_3}(k+2)$ for time $k+2$. Correspondingly, we need to solve the 3-D assignment problem

$$\min_x \sum_{(i,j,m)\in I_3} c_{i,j,m}\rho_{i,j,m}. \tag{4.6.14}$$

where $c_{i,j,m}$ is given in Section 4.4.

In order to get an estimate $\hat{L}$ for the above 3-D assignment problem, we first construct a 2-D assignment problem

$$\min_x \sum_{(i,j,m)\in I_2} c^k_{i,j}\rho_{i,j}, \tag{2D-1}$$

using the track data at time k, and the measurements $z_1(k+1), \ldots, z_{N_2}(k+1)$ for time $k+1$, as described in Section 4.3.

After solving problem (2D-1), we get the (temporary) states of the N_1 tracks at time $k+1$. Using this information, along with the measurements $z_1(k+2), \ldots, z_{N_3}(k+$

2) for time $k+2$, we construct another 2-D assignment problem

$$\min_{x} \sum_{(i,j,m)\in I_2} c_{i,j}^{k} \rho_{i,j}, \tag{2D-2}$$

Suppose the solution to problems (2D-1) and (2D-2) are ρ^1 and ρ^2, respectively. We can get an initial assignment $\hat{\rho}$ as follows:

$$\hat{\rho}_{ijk} = \begin{cases} 1, \text{if } \rho_{ij}^{1} = 1 \text{ and } \rho_{ik}^{2} = 1; \\ 0, \text{ otherwise;} \end{cases} \tag{4.6.15}$$

and get an initial estimate $\hat{L}$ of the total assignment cost by

$$\hat{L} = \sum_{(i,j,m)\in I_3} c_{i,j,m} \hat{\rho}_{i,j,m}. \tag{4.6.16}$$

The complete flow chart of the assignment algorithm is shown in Fig. 4.6.1. The condition (OPT) for 3-D assignment is defined as follows:

$$d_{ijk} \leq 0 \text{ if } \hat{\rho}_{ijk} = 1, \tag{OPT}$$

where $d_{ijk} = c_{ijk} - u_i - v_j - w_k$, as presented in Algorithm 6.

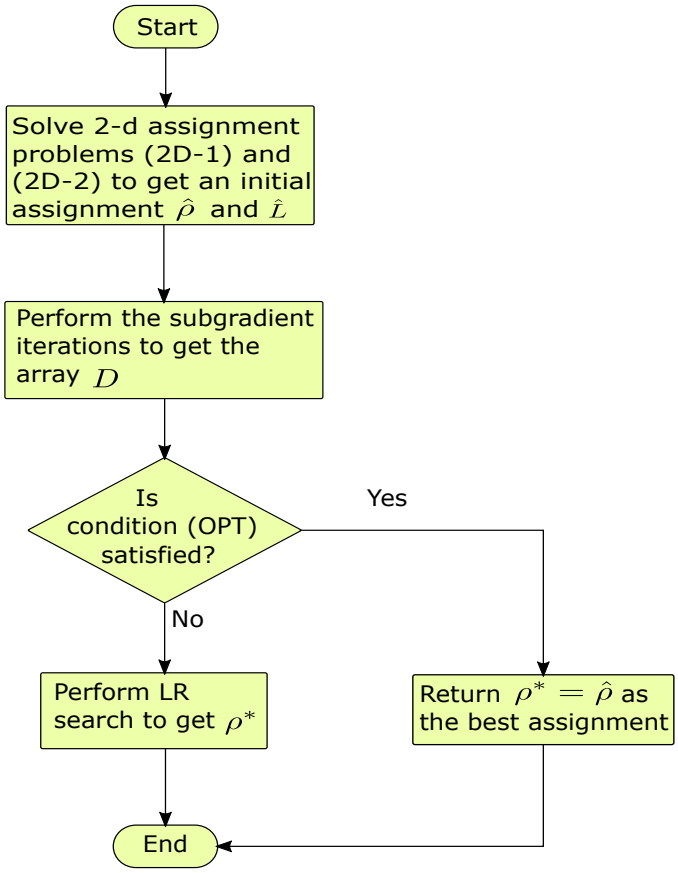

Figure 4.6.1: The complete flowchart of the dual assignment algorithm

4.6.3 Alternating iteration algorithm

A more efficient way to carry out the multidimensional maximization operation can be done by performing a two-stage iterative maximization, with the inner maximization being a two-dimensional assignment problem. Consequently, we get an alternating direction iteration scheme as shown in Algorithm 7 for the 3-dimensional case. The algorithm can be generalized to more general $M-d$ assignment problems in a straightforward manner.

$N_I \leftarrow$ number of iterations;
$I_2 \leftarrow \{(1,2),(1,3),(2,3)\}$;
// External iteration
for $k = 1 : N_I$ **do**
  /* Loop for every 2D combination of the indices */
  for (p,q) *in* I_2 **do**
    $r \leftarrow I_2 \backslash \{p,q\}$ // Set difference /* Solve the following 2D assignment problem */
    $c^{[2]}_{i_p,i_q} = \min_{i_r}\{c_{i_p i_q i_r} - \mu_{r,i_r}\}$
    $(v^*, w^*) \leftarrow \arg\max_{v,w} \sum_{i,j}(c^{[2]}_{ij} - v_i - w_j)^- + \sum_i v_i + \sum_j w_j$
      // $v = (v_1, \ldots, v_N), w = (w_1, \ldots, w_N)$
    $\mu_p \leftarrow v^*$ $\mu_q \leftarrow w^*$
  end
end
return optimal solution (μ_1, μ_2, μ_3) to problem (PD).

Algorithm 7: Alternating direction iteration algorithm

4.7 Simulation results

In this section, we present some simulation results. Our computation environment is a Windows 10 laptop with an Intel i7-6500U CPU. The algorithm is implemented in MATLAB. The proposed algorithm is tested on double frame tracking problems.

Although the state equation (4.2.1) is linear, we have nonlinear measurement equation (4.2.3). The extended Kalman filter is employed in the simulation by linearizing the set of measurement equations:

$$\begin{aligned}\begin{bmatrix} r(k) \\ \theta(k) \end{bmatrix} &= \begin{bmatrix} \sqrt{x^2(k)+y^2(k)} \\ \arctan(\frac{y(k)}{x(k)}) \end{bmatrix} + \begin{bmatrix} w_r(k) \\ w_\theta(k) \end{bmatrix} \\ &\approx H(k) \begin{bmatrix} x(k) \\ y(k) \\ u_x(k) \\ u_y(k) \end{bmatrix} + \begin{bmatrix} w_r(k) \\ w_\theta(k) \end{bmatrix}\end{aligned} \tag{4.7.1}$$

where

$$H(k) = \frac{1}{r(k)} \begin{bmatrix} x(k) & y(k) & 0 & 0 \\ -y(k) & x(k) & 0 & 0 \end{bmatrix} \tag{4.7.2}$$

To assess the performance of the dual assignment algorithm, a series of simulations with different complexities are tested using both the dual algorithm and the Lagrangian relaxation method [33]. For the LR method, the stopping criterion adopted in the simulations is

$$\frac{A_n - G_n}{G_n} < 1\%, \tag{4.7.3}$$

with an upper limit of 200 on the number of iterations, where A_n is the total cost corresponding to the assignment result and G_n is the lower bound estimate at the nth step.

For the dual method, the stopping criterion is $n > 100$ and

$$|L_{n+1} - L_n| < 0.0001 L_n, \tag{4.7.4}$$

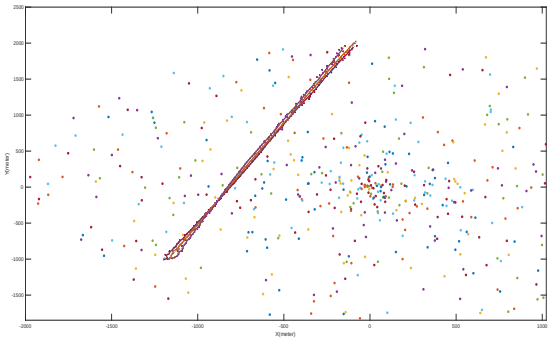

Figure 4.7.1: Three crossing tracks with bearing measurement standard deviations $\sigma_\theta = 0.006$ rad.

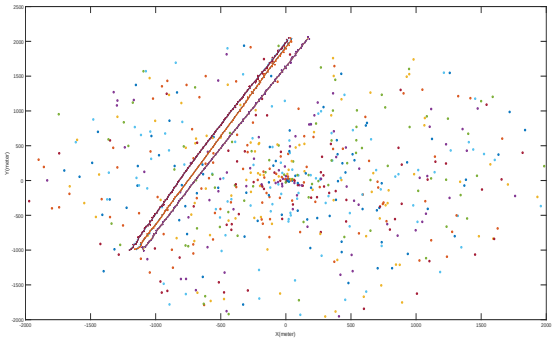

Figure 4.7.2: Three parallel tracks with bearing measurement standard deviations $\sigma_\theta = 0.005$ rad.

with an upper limit of 200 on the number of iterations, where L_n is the lower bound estimation at the nth step.

The results are shown in Fig. 4.7.1 to Fig. 4.7.10. In these figures, the small circles represent false alarms from noises, while the crosses and stars represent measurements from the targets being tracked. The calculated trackes are shown in the figures by solid lines, and the dashed point lines show the ground truths, that is, the true trajectories of the targets. In all these simulations, the noise standard deviations are $\sigma_r = 0.2$ m, $\sigma_x = 0.1\,\mathrm{m/s^2}$, $\sigma_y = 0.1\,\mathrm{m/s^2}$.

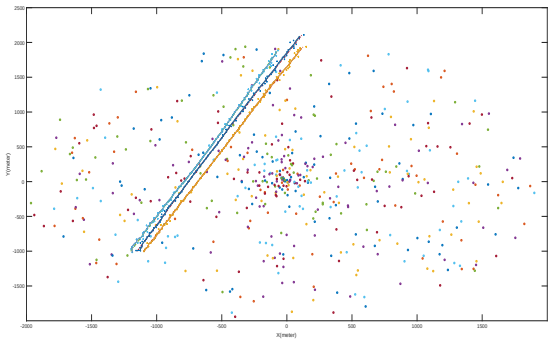

Figure 4.7.3: Three parallel tracks with bearing measurement standard deviations $\sigma_\theta = 0.01$ rad.

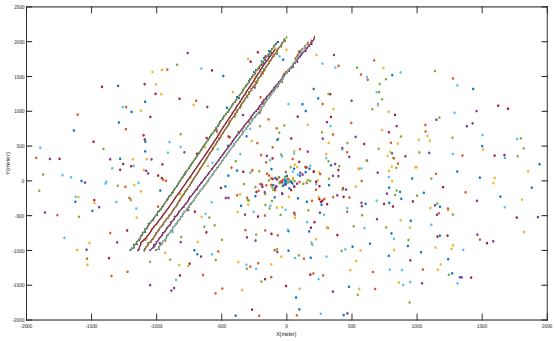

Figure 4.7.4: Five parallel tracks with bearing measurement standard deviations $\sigma_\theta = 0.005$ rad.

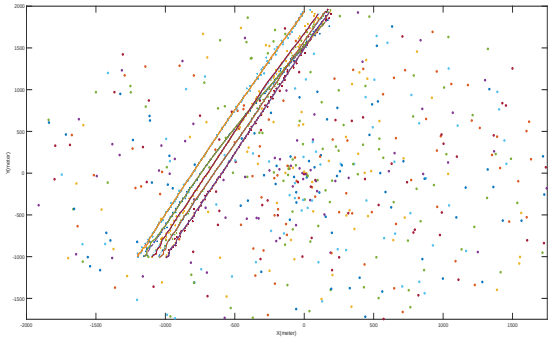

Figure 4.7.5: Five parallel tracks with bearing measurement standard deviations $\sigma_\theta = 0.01$ rad.

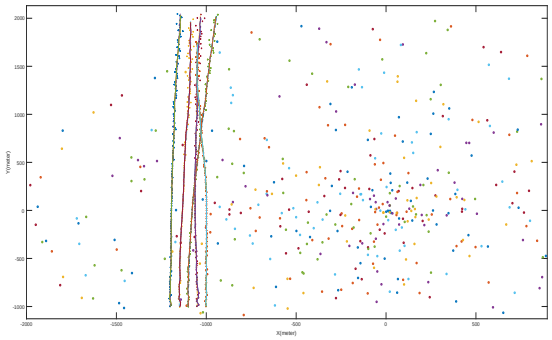

Figure 4.7.6: Five parallel tracks with separation of 50 meters on average, $\sigma_\theta = 0.005$ rad.

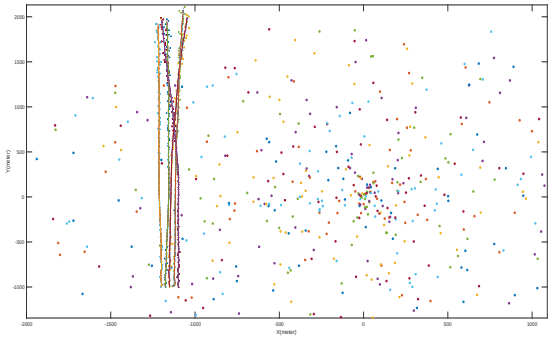

Figure 4.7.7: Five parallel tracks with separation of 25 meters on average, $\sigma_\theta = 0.005$ rad.

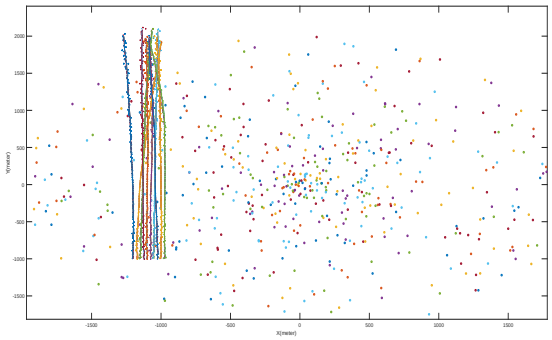

Figure 4.7.8: Ten parallel tracks with separation of 25 meters on average, $\sigma_\theta = 0.005$ rad.

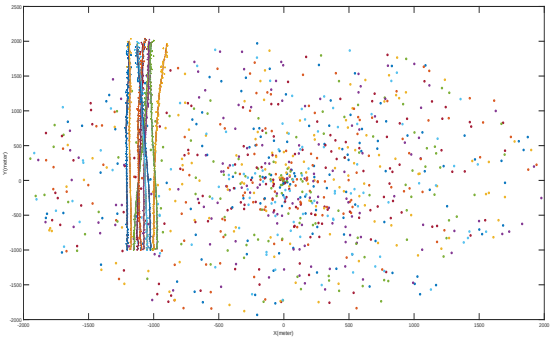

Figure 4.7.9: Ten parallel tracks with separation of 25 meters on average, $\sigma_\theta = 0.005$ rad.

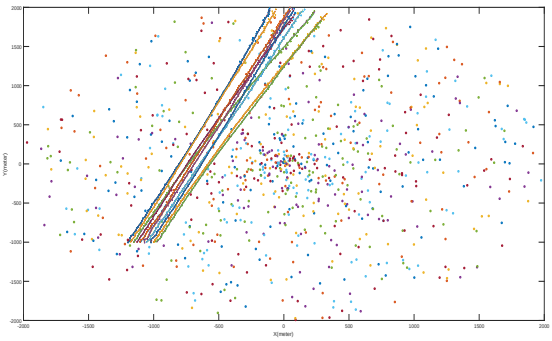

Figure 4.7.10: Ten parallel tracks with separation of 25 meters on average, $\sigma_\theta = 0.005$ rad.

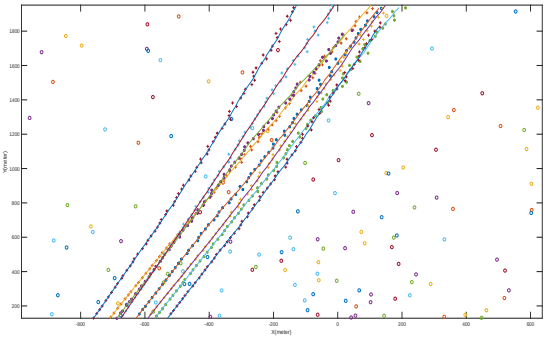

Figure 4.7.11: Eight parallel tracks with small separations.

	Dual method	LR method
Number of successes	97	96
CPU time (s)	1.808504	3.815358
Position RMSE (m)	1.2745	1.2616
Velocity RMSE (m/s)	2.6571	2.6586

Table 4.7.1 Performance comparison between the dual method and the Lagrangian relaxation method based on running 100 Monte Carlo simulations, in each of which 8 targets in nearly parallel movements are tracked for 120 time steps . Here, the number of successes is the number of simulations which give a position RMSE value less than 20 meters, and the CPU time is the time spent on the assignment algorithms for each simulation. The noise standard deviations are $\sigma_r = 0.1$ m, $\sigma_\theta = 0.002$ rad, $\sigma_x = 0.1$ m/s, $\sigma_y = 0.1$ m/s. The initial distance between two neighboring targets is 30 m, the initial velocity is $u_x = 10$ m/s, $u_y = 25$ m/s.

The performance comparison between the dual method and the Lagrangian relaxation method are summarized in Tables 4.7.1 to 4.7.3. These comparisons are based on the scenario illustrated in Fig. 4.7.11, with several different parameters. The results demonstrate that the dual approach outperforms the LR algorithm in terms of the elapsed CPU times, and both methods have basically similar accuracy in terms of the RMSE values.

	Dual method	LR method
Number of successes	98	77
CPU time (s)	1.453611	3.844651
Position RMSE (m)	2.7263	2.7470
Velocity RMSE (m/s)	2.7931	2.7927

Table 4.7.2 Performance comparison between the dual method and the Lagrangian relaxation method based on running 100 Monte Carlo simulations, in each of which 8 targets in nearly parallel movements are tracked for 120 time steps . Here, the number of successes is the number of simulations which give a position RMSE value less than 20 meters, and the CPU time is the time spent on the assignment algorithms for each simulation. The noise standard deviations are $\sigma_r = 0.2$ m, $\sigma_\theta = 0.005$ rad, $\sigma_x = 0.1$ m/s, $\sigma_y = 0.1$ m/s. The initial distance between two neighboring targets is 30 m, the initial velocity is $u_x = 10$ m/s, $u_y = 25$ m/s.

	Dual method	LR method
Number of successes	98	99
CPU time (s)	1.513820	3.907298
Position RMSE (m)	2.6706	2.6688
Velocity RMSE (m/s)	2.7911	2.7970

Table 4.7.3 Performance comparison between the dual method and the Lagrangian relaxation method based on running 100 Monte Carlo simulations, in each of which 8 targets in nearly parallel movements are tracked for 120 time steps . Here, the number of successes is the number of simulations which give a position RMSE value less than 20 meters, and the CPU time is the time spent on the assignment algorithms for each simulation. The noise standard deviations are $\sigma_r = 0.2$ m, $\sigma_\theta = 0.005$ rad, $\sigma_x = 0.1$ m/s, $\sigma_y = 0.1$ m/s. The initial distance between two neighboring targets is 50 m, the initial velocity is $u_x = 10$ m/s, $u_y = 25$ m/s.

4.8 Conclusions

In this chapter, we formulated the data association problem in multi-frame tracking as multidimensional assignment problems, and applied the dual assignment algorithm developed in the previous two chapters to compute the resulting assignment problems. Simulations on a series of scenarios with different complexities were conducted to test the performance of the newly proposed assignment algorithms. The results showed that the dual assignment algorithm is faster than the Lagrangian method, while demonstrating the same level of accuracy.

Chapter 5

Conclusions and future work

5.1 Summary

In this thesis, we developed a novel assignment algorithm based on the idea of dual convexification. We first developed the algorithm in three-dimensional cases. A convex dual formulation was presented, then we showed that the dual formulation is equivalent to the Lagrangian relaxation method in terms of their accuracy of approximating the optimal assignment value. Despite the above equivalence, however, the dual formulation is much simpler and elegant than the Lagrangian relaxation method. As a result, it was shown in the simulations on a set of random generated assignment problems with sizes varying from 10x10x10 to 100x100x100, the dual algorithm is more efficient than the Lagrangian relaxation method. Meanwhile, due to the simplicity of the dual formulation, the newly proposed algorithm is much easier to implement than the Lagrangian relaxation method.

Next, the dual algorithm was extended to handle more general multidimensional assignment problems. Moreover, an effective local searching method was developed to

further improve the accuracy. The results of simulations on four-dimensional assignment problems with different sizes demonstrated that the dual algorithm outperforms the Lagrangian method.

The newly proposed algorithm was further applied to solve assignment problems arising from a series of multi-frame tracking problems with different complexities. The simulation results showed that the dual algorithm is more efficient than the Lagrangian relaxation method in this context, while maintaining the same level of high accuracy. We conclude that the dual method will provide an efficient and effective solution to the data association problems in multi-target multi-sensor or multi-frame tracking applications.

5.2 Contributions made by the thesis

- Proposed a purely dual formulation for the multi-dimensional assignment problem and obtained the conditions under which the duality gap reaches zero, and thus provided a theoretical framework for analyzing the assignment problem.

- Established the equivalence between the dual formulation and the Lagrangian relaxation method in the sense that both provide the same lower bound on the optimal value of the assignment problem, and thus obtained a sufficient and necessary condition for the Lagrangian method to get the optimal assignment value with guarantee.

- Stated that the Lagrangian relaxation procedure essentially amounts to relaxing the binary constraint of the original assignment problem, and thus equivalent to solving a Linear programming (LP) problem together with a search procedure

to recover the binary constraint.

- Demonstrated that dual formulation provides a streamlined and efficient solution procedure for solving the multi-dimensional assignment problem.
- Developed a systematic algorithm for applying the dual algorithm to multi-frame tracking problems.

5.3 Future work

The dual algorithm will be extended further to solve more general one-to-many and many-to-many assignment problems, in order to handle the data associations arising from extended target tracking applications.

On the other hand, an m-best algorithm based on the dual formulation will be developed, to satisfy the requirement for ranking several different possible hypotheses in the MHT algorithm.

References

[1] F. Abdallah, G. Nassreddine, and T. Denœux. A multiple-hypothesis map-matching method suitable for weighted and box-shaped state estimation for localization. *IEEE Transactions on Intelligent Transportation Systems*, 12:1495–1510, 2011.

[2] M.L. Balinski. Signature methods for the assignment problem. *Operations Research*, 33:527–536, 1985.

[3] H.J. Bandelt, Y. Crama, and F.C. Spieksma. Approximation algorithms for multi-dimensional assignment problems with decomposable costs. *Discrete Applied Mathematics*, 49:25–50, 1994.

[4] Y. Bar-Shalom, P.K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, Storrs, CT, 2011.

[5] D.P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, Belmont, Massachusetts, 2003.

[6] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 2003.

[7] D.P. Bertsekas and D.A. Castanon. A forward/reverse auction algorithm for asymmetric assignment problems. *Computational Optimization and Applications*, 1:277–279, 1992.

[8] S.S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 19:5–18, 2004.

[9] R.E. Burkard. Selected topics on assignment problems. *Discrete Applied Mathematics*, 123:257–302, 2002.

[10] P.M. Camerini, L. Fratta, and F. Maffioli. On improving relaxation methods by modified gradient techniques. *Mathematical Programming Study*, 3:26–34, 1975.

[11] S. Deb, M. Yeddanapudi, K.R. Pattipati, and Y. Bar-Shalom. A generalized s-d assignment algorithm for multisensor-multitarget state estimation. 33:523–538, 1997.

[12] M. Dell'Amicoa and P. Tothb. Algorithms and codes for dense assignment problems: The state of the art. *Discrete Applied Mathematics*, 100:17–48, 2000.

[13] P. Emami, P. M. Pardalos, L. Elefteriadou, and S. Ranka. Machine learning methods for solving assignment problems in multi-target tracking. *arXiv preprint, arXiv:1802.06897*, 2018.

[14] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons Ltd, Chichester, 1987.

[15] A.M. Frieze and J. J. Yadegar. An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice. *Journal of the Operational Research Society*, 32:989–995, 1981.

[16] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, CA, 1979.

[17] V. Garg and T. Wickramarathne. Mht approach to ubiquitous monitoring of spatio-temporal phenomena. *21st International Conference on Information Fusion, Cambridge, UK*, pages 1816–1821, 2018.

[18] D.A. Grundel and P.M. Pardalos. Test problem generator for the multidimensional assignment problem. *Computational Optimization and Applications*, 30:133–146, 2005.

[19] Sherali H.D. and C.H. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7:1–31, 1995.

[20] V. Jeyakumar, G.M. Lee, and S. Srisatkunarajah. New kuhn–tucker sufficiency for global optimality via convexification. *Nonlinear Analysis*, 71:373–381, 2009.

[21] R. Jonker and Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

[22] D. Karapetyan and G. Gutin. Local search heuristics for the multidimensional assignment problem. *Journal of Heuristics*, 17:201–249, 2011.

[23] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[24] T. Kirubarajan, Y. Bar-Shalom, and K.R. Pattipati. Multiassignment for tracking a large number of overlapping objects. *IEEE Transactions on Aerospace and Electronic Systems*, 37:2–21, 2001.

[25] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, and I. Kadar. Ground target tracking with variable structure imm estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36:26–46, 2000.

[26] H.W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[27] R.A. Lau and J.L. Williams. Multidimensional assignment by dual decomposition. *Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Adelaide, Australia*, pages 437–442, 2011.

[28] E.L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14:699–719, 1966.

[29] D. Li. pth power lagrangian method for integer programming. *Annals of Operations Research*, 98:151–170, 2000.

[30] J. Li, Kirubarajan T., R. Tharmarasa, Brown D., and K. R. Pattipati. A dual approach to multi-dimensional assignment problems. 2019. submitted to Journal of Global Optimization, June.

[31] J. Li, R. Tharmarasa, D. Brown, T. Kirubarajan, and K. R. Pattipati. A novel convex dual approach to three-dimensional assignment problem: theoretical analysis. *Computational Optimization and Applications*, 2019. https://doi.org/10.1007/s10589-019-00113-w.

[32] T. Mizuike and Y. Ito. Optimization of frequency assignment. *IEEE Transactions on Communications*, 37:1031–1041, 1989.

[33] K.R. Pattipati, S. Deb, Y. Bar-Shalom, and R.B. Washburn. A new relaxation algorithm and passive sensor data association. *IEEE Transactions on Automatic Control*, 37:198–212, 1992.

[34] D.W Pentico. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176:774–793, 2007.

[35] S. Plitsos, D. Magos, and I. Mourtos. An integrated solver for multi-index assignment. *International Symposium on Artificial Intelligence and Mathematics*, 2016.

[36] A. B. Poore and S. Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modelling*, 43:1074–1091, 2006.

[37] A.B. Poore and N. Rijavec. A lagrangian relaxation algorithm for multidimensional assignment problems arsing from multitarget tracking. *SIAM Journal on Optimization*, 3:544–563, 1993.

[38] A.B. Poore and A. J. Robertson III. A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems. *Computational Optimization and Applications*, 8:129–150, 1997.

[39] A.J. Robertson. A set of greedy randomized adaptive local search procedure (grasp) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19:145–164, 2001.

[40] G. Terry Ross and Richard M. Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, 8:91–103, 1975.

[41] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2:121–152, 1992.

[42] N.Z. Shor. *Minimization Methods for Non-differentiable Functions*. Springer-Verlag, Berlin, 1985.

[43] M.I. Skolnik. *Introduction to radar systems*. The McGraw-Hill Companies, Inc, 2001.

[44] C. Vogiatzis, E. L. Pasiliao, and P. M. Pardalos. Graph partitions for the multidimensional assignment problem. *Computational Optimization and Applications*, 58:205–224, 2014.

[45] J. L. Walteros, C. Vogiatzis, E. L. Pasiliao, and P. M. Pardalos. Integer programming models for the multidimensional assignment problem with star costs. *European Journal of Operational Research*, 235:553–568, 2014.

[46] H. Wang, T. Kirubarajan, and Y. Bar-Shalom. Large scale air traffic surveillance using imm estimators with assignment. *IEEE Transactions on Aerospace and Electronic Systems*, 35:255–266, 1999.