# Image Segmentation via Spectral Clustering with Density Sensitive Similarity Function

**Xuecheng Zhang**
The Australian National University
Canberra, Australia
u6284513@anu.edu.au

## Abstract

Image segmentation is the categorisation of an image into various subgroups. K-means algorithm is one of the most conventional clustering algorithms used for image segmentation. However, k-means clustering algorithm does not excel at non-linear classification. Thus, we introduce spectral clustering (SC) to deal with complex and non-linear models. In this paper, we start with k-means++ algorithm. Then we introduce two spectral clustering algorithms – normalised-cut (Ncut) and spectral clustering via density sensitive similarity (DSSC) algorithm and compare their performance with different metrics. Experimental results demonstrate that DSSC surpasses other models on the CIFAR-10 dataset on each of metrics.

## 1 Introduction

Image segmentation separates an image into various sections, each with its own set of properties. Previously, the image segmentation methods were used in the bio-medical area, where specific targets in an image were segmented and then analysed for medical diagnosis. Due to the simplicity of the medical image and the clear distinction between the background and the target, most of the image segmentation methods were segmented by simple threshold-based methods [1]. Recently, image segmentation is widely applied in areas such as intelligent security, unmanned vehicles, satellite remote sensing, medical image processing and biometrics.

The most intuitive way to address image segmentation is to rely on naive k-means clustering to split an image into k pieces. However, the naive k-means algorithm aims to solve convex objectives and it cannot deal with sophisticated images in real life. Thus, spectral clustering (SC) is purposed to address the non-convex problems. The idea of creating graph partitions based on the eigenvectors of the adjacency matrix was first proposed by Donath and Hoffman in 1973 [2]. In the same year, Fiedler discovered that the second smallest eigenvector can be used to partition the graph [3]. SC has been discovered and extended by 1990s [4]. Shi and Malik purposed a Ncut criteria for measuring total dissimilarity across clusters as well as total similarity within clusters [5]. Afterwards, Ng et al. [6] purposes a new spectral clustering approach based on matrix perturbation theory and they exhibit promising experimental results on a variety of difficult clustering challenges.

However, images may contain multiple scales in real life and using Euclidean distance alone cannot obtain the distribution of data, resulting in deficient clustering performance. Hence, we extend the SC algorithm to replace the Gaussian similarity function with density sensitive similarity algorithm [7]. In this paper, we initially implement k-means++ and then implement two SCs- one based on [6] and the other is based on density sensitive similarity (i.e. DSSC algorithm [7]). Finally, we compare their performance among different metrics on CIFAR-10 dataset.

We highlight the main contributions of this paper as below:

1. We have successfully replicated k-means++, SC algorithm on image segmentation tasks.

2. In previous works [4–6], they analyse the result by qualitative results alone. In this paper, we measure their performance by not only qualitative results but also quantitative results. We compare their performances through different metrics – accuracy (ACC), F-score (F1-score) and normalized mutual information (NMI). The experimental results demonstrate that DSSC exceeds other algorithms on CIFAR-10 dataset.

# 2 Problem Formulation

## 2.1 Problem Definition

According to [1], given an image $\boldsymbol{X}$ and a homogeneity predicate $P(\cdot)$ defined on clusters of connected pixels, image segmentation is to divide the image $\boldsymbol{X}$ into disjoint non-empty subsets (regions) $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_N$, such that:

$$\bigcup_{i=1}^{N} \boldsymbol{c}_i = \boldsymbol{X} \tag{1}$$

$$c_i \bigcap c_j = \text{ where } i \neq j \tag{2}$$

$$P(c_i) = \text{TRUE } \forall i \in \{1, 2, \cdots, N\} \tag{3}$$

$$P(c_i \cup c_j) = \text{FALSE where } i \neq j \tag{4}$$

The (3) illustrates that each pixels in the same region possess similar properties. The (4) states that pixels in adjacent region possess distinct properties. In other words, the image segmentation problem can be regarded as group the pixels according to the *similarity* of data features.

## 2.2 K-means++ Clustering

K-means algorithm can regard pixels as separate data points. Assume we have a D-dimensional data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, we denote the centres of clusters as $\mu_k$ $(k = 1, 2, \cdots, K)$ and our objective is to minimise the sum of the squares of the distances (often using Euclidean distances) from all samples to their nearest cluster centres is minimised. Such an objective function attempts to impel data to be similar in a cluster. We can define the loss function J [8] where binary indicator $\gamma_{nk}$ describes whether the data is $\boldsymbol{x}_n$ is assigned to cluster centre $k$:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||_2^2 \tag{5}$$

The clustering problem converts to find optimal values for $\{\gamma_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$ to minimise $J$ loss function. In section 3.1, we will demonstrate how to minimise $J$ using EM algorithm.

## 2.3 Ncut

In SC, they regard the coloured image $\boldsymbol{X}$ as the graph $G = (V, E)$ and each vertex $v_i$ stands for a pixel $x_i$ in the coloured image. Intuitively, the definition of exemplary image segmentation is that pixels which are assigned to an identical cluster should be particularly similar and pixels which are assigned to a distinct cluster should be particularly dissimilar. The clustering problem may be restated using the similarity graph: we want to partition the image so that the similarity (usually measured by Euclidean distance) within a group is high and the similarity across the group is low.

We introduce the graph-cut approach to address the clustering issue. The graph-cut approach quantifies the degree of dissimilarity between these two groups using the total weight of the edges that have been disconnected. We can start with considering a simple bi-partitional cut. Given a partition of image $\boldsymbol{X}$ into two sets A and B. The value of cut is defined as (6).

$$cut(A, B) = \sum_{i \in A, j \in B} W(i, j) \tag{6}$$

$$W(i, j) = \begin{cases} e^{-\frac{||\boldsymbol{F}(i) - \boldsymbol{F}(j)||_2^2}{\sigma_I^2}} * e^{-\frac{||\boldsymbol{X}(i) - \boldsymbol{X}(j)||_2^2}{\sigma_X^2}} & \text{if } ||X(i) - X(j)||_2 < r \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

W(•,•) is the weight of the edge between two vertices and the weight represents the degree of similarity. [5] defines the similarity graph as (7) in which $F(\bullet)$ presents the brightness that is the average of RGB channel, $X(\bullet)$ represents the spatial position. Additionally, $\sigma_I$, $\sigma_X$ are for normalisation, particularly $\sigma_I = p_{ratio}(\max||F(i) - F(j)||_2 - \min||F(i) - F(j)||_2)$, $\sigma_X = p_{ratio}(\max||X(i) - X(j)||_2 - \min||X(i) - X(j)||_2)$. Shi and Malik [5] purposed the Ncut (8) and they take each vertex in the graph into consideration to prevent the unbalanced cut.

$$Ncut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)} \tag{8}$$

where $vol(A) = \sum_{i \in A} W(i, j)$. When we expand the bi-partitional cut to k-way Ncut, it can be defined as minimising (9):

$$Ncut_k = \frac{cut(A_1, V - A_1)}{vol(A_1)} + \cdots + \frac{cut(A_k, V - A_k)}{vol(A_k)} \text{ where } V = \bigcup_{i=1}^{k} A_i \tag{9}$$

In the section 3.2, we will demonstrate how to find the minimum k-way Ncut.

### 2.4 DSSC

The traditional Euclidean distance only reflects the local consistency of the clustering structure, but not the global consistency. For example in Figure. 1, although point E and point A are in the same cluster and at the same time, point A and point F are in the different clusters, we obtain that point E has a lower probability in the same cluster as point A than point F according to Euclidean distance. That is to say, we need a new distance function which does not guarantee that the direct path between two points is not necessarily the shortest. In other words, the length of a path through a region of high density connected by a shorter edge is shorter than the distance between two points connected directly through a region of low density, i.e. $d(A, B) + d(B, C) + d(C, D) + d(D, E) < d(A, E)$ where d(•,•) denotes the distance between two distinct points.

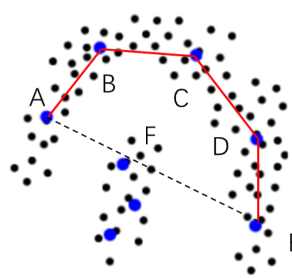



Figure 1: According to the shortest distance, point F is more likely to be classified in the same cluster as point A than point E. (since $d(A, F) < d(A, E)$)

Yang et al. [7] suggest that the adjustable line segment length as below:

$$L(\boldsymbol{x}_i, \boldsymbol{x}_j) = (e^{\rho dist(\boldsymbol{x}_i, \boldsymbol{x}_j)} - 1)^{\frac{1}{\rho}} \tag{10}$$

where $\boldsymbol{x}_i$,$\boldsymbol{x}_j$ are data which consists of colour information, $dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, ρ is the density factor which is greater than 1. Since the distance function violates the triangle inequality, the distance function cannot be straightforwardly utilised as a similarity matrix. Consider $S = \{s_1, \cdots, s_l\} \in V$ to be the path $s_1$ to $s_l$ with length $l = |S|$, where edges ($s_k$, $s_k + 1) \in E$, $1 \leq$ k $< l$. Let $S_{ij}$ ($1 \leq i$, $j \leq n$) denote the set of all paths connecting with the data points $x_i$, $x_j$. According to [7], the density-sensitive distance between data points to $x_i$, $x_j$ is defined as:

$$D_{ij} = min_{s \in S_{ij}} \sum_{k=1}^{l-1} L(s_k, s_{k+1}) \tag{11}$$

Intuitively the further the distance between two data points, the less they are similar. Thus, the similarity matrix can be viewed as the inverse of the distance D. After we obtain the similarity

$$W_{ij} = \frac{1}{1 + D_{ij}} \tag{12}$$

matrix, we minimise function (9) and obtain the optimal k-way Ncut similar to section 2.3.

# 3 Methods

## 3.1 K-means++ Clustering

The main idea of the k-means initialisation is to locate each cluster centroid as far away as possible. Then, in the expectation phase, we minimise $J$ with respect to $\gamma_{nk}$ while fixing $\boldsymbol{\mu}_k$. Afterwards, in the maximisation phase, we minimise $J$ with respect to $\boldsymbol{\mu}_k$ [8] while fixing $\gamma_{nk}$. These two phases are iterated until convergence. For expectation phase, we can rewrite $\gamma_{nk}$ as

$$\gamma_{nk} = \begin{cases} 1 & \text{if } k = argmin_j ||\boldsymbol{x}_n - \boldsymbol{\mu}_j||_2^2 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

In the second phase, we obtain the expected $\gamma_{nk}$. We can calculate the partial derivative $\frac{\partial J}{\partial \gamma_{nk}}$:

$$2\sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0 \tag{14}$$

$$\Longrightarrow \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n \tag{15}$$

**Algorithm 1:** Kmeans++ clustering algorithm

**input :** $\boldsymbol{X}$: $x_1, x_2, \cdots, x_N$, #centroids: $k$, #iterations: $max\_iter$
**output:** Label $\boldsymbol{\gamma}$ and clustering centre $\boldsymbol{\mu}$

**init**
- Choose a point uniformly from $\boldsymbol{X}$ as the first clustering centre $\mu_1$.
- **while** *len*($\boldsymbol{\mu}$) $< k$ **do**
  - Choose another clustering centre $\mu_i$ from $\boldsymbol{X}$ with the probability $\frac{D(x)^p}{\sum_{x \in \boldsymbol{X}} D(x)^p}$

**for** $iter$ **to** $max\_iter$ **do**
- **for** *i* **to** *N* **do**
  - $\gamma_i \leftarrow argmin_j ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||_2^2$
- **for** *j* **to** *K* **do**
  - $\mu_j \leftarrow \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$

## 3.2 Ncut

In order to address the problem, we can start with considering the simple bi-partional cut. Conforming to [5], they introduce a indicator $\boldsymbol{x}$. When $\boldsymbol{x}_i = 1$, the vertex $i$ belongs to set A and -1, otherwise. (8) can be rewritten as below:

$$min \frac{\sum_{(\boldsymbol{x}_i > 0, \boldsymbol{x}_j < 0)} -\boldsymbol{W}_{ij} \boldsymbol{x}_i \boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i > 0} \boldsymbol{D}_{ii}} + \frac{\sum_{(\boldsymbol{x}_i < 0, \boldsymbol{x}_j > 0)} -\boldsymbol{W}_{ij} \boldsymbol{x}_i \boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i < 0} \boldsymbol{D}_{ii}} \tag{16}$$

As stated by [5], after simplification of (8), we are able to obtain as below:

$$min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T (\boldsymbol{D} - \boldsymbol{W}) \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} \text{ where } y = (\boldsymbol{1} + \boldsymbol{x}) - b(\boldsymbol{1} - \boldsymbol{x}). \tag{17}$$

According to Raleigh Theorem [5], we can conclude that the minimum value of (17) is given by the second smallest eigenvalues of the Laplacian $\boldsymbol{L}$. In this paper, the normalised Laplacian L is defined as $L_{sym} = D^{-1/2}(D - W)D^{-1/2}$. Since the next few eigenvectors may contain advantageous information for segmentation. When we expand the bi-partional cut to k-way Ncuts, we can use k-means clustering algorithm on these eigenvectors $\mathbb{R}^k$ to cluster these pixels in $c_1, c_2, \cdots, c_k$.

**Algorithm 2:** Ncut algorithm

**input :** $\boldsymbol{X}$: $x_1, x_2, \cdots, x_N$, #centroids: $k$, #iterations: $max\_iter$
**output:** Image segmentation $\boldsymbol{X}$

1 Construct the weighted adjacency matrix W using (7).
2 Compute the normalised Laplacian $L_{sym} = D^{-1/2}(D - W)D^{-1/2}$.
3 Compute the first $k$ eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k$ of $L_{sym}$.
4 Scale $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k$ to unit length i.e. $v_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$.
5 Cluster the points $V : v_1, v_2, \cdots, v_n$ with k-means++ algorithm 1 to generate $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$.
6 **for** $i$ **to** $k$ **do**
7     Find all indices $idx_1, idx_2, \cdots, idx_{last}$ which belong to class $i$
8     Calculate the mean of $X_{idx_1}, X_{idx_2}, \cdots, X_{idx_{last}}$ as the new image pixel for every $X_{idx_1}, X_{idx_2}, \cdots, X_{idx_{last}}$.

### 3.3 DSSC

At first, DSSC uses k-nearest neighbours (KNN) to construct the neighbourhood similarity graph so that each vertex will be connected to at least k vertices. Then, we apply Dijkstra algorithm to $D_{ij}$ to calculate the shortest path between two vertices $v_i$ and $v_j$. Afterwards, the similarity matrix is obtained by (12). After obtaining the similarity matrix, the subsequent step is identical to Ncut algorithm.

**Algorithm 3:** DSSC algorithm

**input :** $\boldsymbol{X}$: $x_1, x_2, \cdots, x_N$, #centroids: $k$, #iterations: $max\_iter$
**output:** Image segmentation $\boldsymbol{X}$

1 Use KNN to construct k-nearest neighbour graph G.
2 Apply Dijkstra algorithm to the graph G and obtain density sensitive distance metric $\boldsymbol{D}$
3 Construct the weighted adjacency matrix $\boldsymbol{W}$ using (12).
4 Identical to the step 2-8 in algorithm 2

## 4 Experiments

We conduct the experiments on the horse, deer and airplane images from CIFAR-10 dataset. Since, there is no ground truth for segmentation in CIFAR-10 dataset, we manually segment 10 images for each of these three labels and we perform image segmentation on these 10 images. In the pre-processing step, we normalise each pixel value to [0,1].

### 4.1 Image Segmentation: The CIFAR-10 Dataset

We compare four distinct models, settings of which are described below:

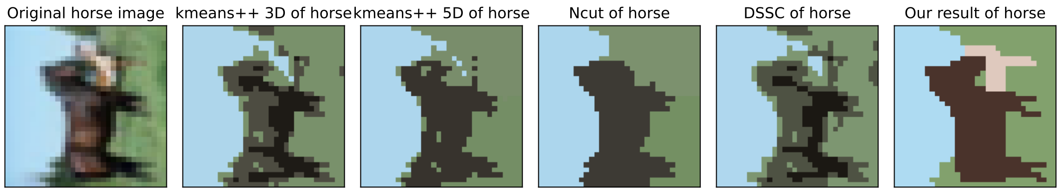
**K-means++ 3D clustering:** We consider the 3D pixel (R,G,B) in the coloured image as the input $\boldsymbol{X}$ of k-means algorithm.

**K-means++ 5D clustering:** This is similar to the previous model, the only difference being the addition of spatial information to the 3D pixel. Before performing k-means++ algorithm, we normalise the 5D data. In the end, we de-normalise the result of k-means++ to obtain the result of image segmentation.

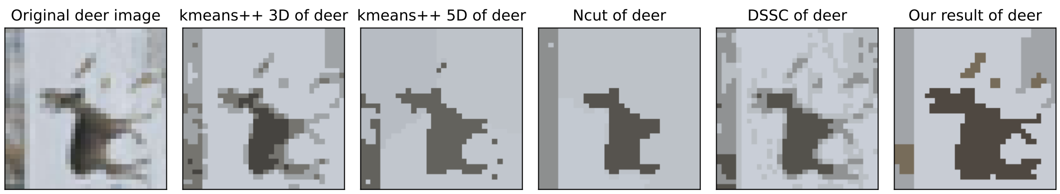
**Ncut:** We set the ratio $p_{ratio}$ and $r$ to 0.098 and 16 respectively.

**DSSC:** We set the density factor ρ and the number of nearest neighbours $k$ are 20 and 19 respectively.

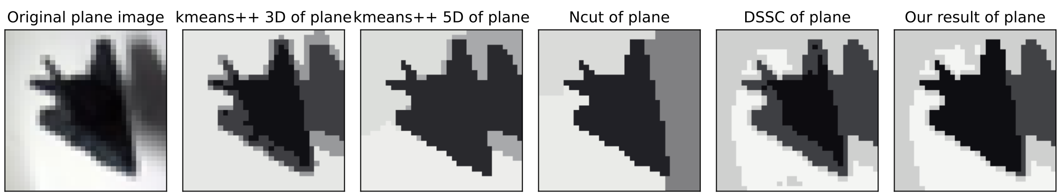
Note that among all of the models, we use $D^4$ weight (i.e. $\frac{D(x)^4}{\sum_{x \in \boldsymbol{X}} D(x)^4}$) in kmeans++ initialisation step. In addition, $k$ and $max\_iter$ are set to 4 and 60 respectively.




(a) The image segmentation of a horse image




(b) The image segmentation of a deer image




(c) The image segmentation of an airplane image

Figure 2: The comparison of k-means++, Ncut and DSSC algorithm on horse, deer, airplane images

As shown on Figure 2, the DSSC algorithm not only performs a decent work in image segmentation, but also demonstrates more details than the other models. For instance, DSSC algorithm is able to locate the feet of deer (Figure 2(b)) and capture the light and shadow of horse and airplane (Figure 2(a),2(c)). The DSSC utilises density-sensitive similarity measurements, which adjust the similarity in dense area than Gaussian kernel. Thus it is more likely to discover data with different scales. On the contrary, Ncut and kmeans 5D are inferior. Their image segmentation result is lacking the information on deer's feet. (Figure 2(b)).

### 4.2 Evaluation Metrics

The performance of SC can be evaluated by the quantitative result. In this paper, we use three common clustering metrics, each of which is described below:

**ACC:** It compares the classes between true labels $y$ and predicted labels $\hat{y}$.

**F-score:** is the harmonic mean of precision and recall and can be utilised to determine the performance and robustness of the model.

$$\text{precision} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{19}$$

$$\text{F-score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{20}$$

**NMI:** NMI determines the similarity (based on information entropy) between the true labels $y$ and the predicted labels $\hat{y}$. When $y$ and $\hat{y}$ obtain similar information, the numerator of NMI possesses large value. Additionally, the denominator of NMI normalises the formula to [0,1].

$$NMI = \frac{2I(y; \hat{y})}{H(y) + H(\hat{y})} \tag{21}$$

Before evaluation, we use Hungarian method [9] to find the mapping between $y$ and $\hat{y}$. After obtaining a perfect mapping with $y$ and $\hat{y}$, we evaluate the performance of k-means 3D, k-means 5D, SC and DSSC algorithms based on accuracy, F-score and NMI metrics.

Table 1: Overall ACC, F-score and NMI of k-means++ 3D/5D, SC and DSSC over 30 runs

| Algorithm | ACC | F-score | NMI |
|---|---|---|---|
| K-means++ 3D | 0.69 | 0.72 | 0.54 |
| K-means++ 5D | 0.63 | 0.57 | 0.48 |
| Ncut | 0.71 | 0.66 | 0.52 |
| **DSSC** | **0.76** | **0.75** | **0.62** |

From Table 1, we can see that DSSC outperforms the other algorithms in all metrics and the difference is statistically significant ($p < 0.05$ in all cases, t-test). Moreover, DSSC algorithm scores higher than Ncut for each of the evaluation criteria indicating that using density sensitive similarity function instead of Gaussian function as the similarity matrix improves significantly in image segmentation. The quantitative result matches with qualitative result in Figure 2.

## 5 Conclusion

In this paper, we implement k-means 3D/5D, Ncut and DSSC algorithms and compare these algorithms on image segmentation problems on CIFAR-10 dataset with ACC, F-score and NMI metrics. From qualitative results, we demonstrate that the density sensitive similarity based similarity matrix capture information about different scales. From quantitative result, we demonstrate that spectral clustering with density sensitive similarity function outperforms over the other models obtaining 0.76 score in ACC, 0.75 score in F-score and 0.62 in NMI. We believe that it can be useful to many future image segmentation projects.

### References


[1] King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981.

[2] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[3] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[4] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[5] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[6] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[7] Peng Yang, Qingsheng Zhu, and Biao Huang. Spectral clustering with density sensitive similarity function. *Knowledge-Based Systems*, 24(5):621–628, 2011.

[8] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[9] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.