



## Simulation du problème du voyageur de commerce pour la distribution de fruits dans la ville de Bogor avec la méthode du recuit simulé.

Ali Khumaidi<sup>1</sup>, Ridwan Raafi'udin<sup>2</sup>, Indra Permana Solihin<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika,

Universitas Krisnadwipayana, Jl. Kampus Unkris, Jatiwaringin, Bekasi, Indonésie

<sup>2,3</sup> Universitas Pembangunan Nasional Veteran Jakarta, Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, Indonésie

E-mail : [alikhumaidi@unkris.ac.id](mailto:alikhumaidi@unkris.ac.id), [raafiudin@upnvj.ac.id](mailto:raafiudin@upnvj.ac.id), [indrapermanasolihin@upnvj.ac.id](mailto:indrapermanasolihin@upnvj.ac.id)

### INFO ARTICLE

Historique des articles : Reçu : 25/01/2020 Révisé : 01/02/2020 Accepté : 13/02/2020

#### Mots-clés :

Problème du voyageur de commerce, TSP, recuit simulé, distribution des fruits, itération.

### RÉSUMÉ

Le problème du vendeur itinérant (TSP) est un problème consistant à trouver la distance la plus courte lorsqu'un vendeur visite un certain nombre de villes, à condition que chaque ville soit visitée exactement une fois et qu'il retourne ensuite à la ville initiale. Simulations de TSP pour la distribution de fruits dans la ville de Bogor, chaque emplacement ayant des coordonnées  $x$  et  $y$  comme distances. La TSP utilisée est une TSP symétrique dont la distance de la ville  $a$  à  $b$  a la même distance que la ville  $b$  à  $a$ . Résoudre et trouver des solutions à des problèmes en utilisant l'algorithme du recuit simulé (SA). Le principe de fonctionnement est le suivant : à haute température, les particules de métal liquide ont un niveau d'énergie élevé, il est donc relativement facile de se déplacer contre d'autres particules. Puis, lorsque la température baisse, la particule s'ajuste lentement pour former une configuration de manière à obtenir un état stable avec un niveau d'énergie minimum. Cette énergie minimale correspond à la distance la plus courte. D'après les expériences qui ont été faites en utilisant l'algorithme SA sur le problème TSP, les résultats montrent que le nombre d'itérations qui produisent la solution optimale dépend du nombre d'emplacements simulés. Plus le nombre de points de localisation simulés est élevé, plus le nombre d'itérations nécessaires est important.

Copyright © 2020 Jurnal Mantik.  
Tous droits réservés.

### 1. Introduction

Les problèmes de livraison des fruits et légumes aux commerçants du marché sont souvent dus à une réception tardive. En outre, d'autres problèmes sont dus à l'efficacité du temps et des coûts, de sorte que la précision est nécessaire pour déterminer le chemin ou la route la plus courte entre les lieux. Parce qu'il y a des problèmes dans la détermination de l'efficacité du temps et du coût dans la détermination du chemin ou de la route la plus courte, il est nécessaire de trouver une solution pour les surmonter. Avec le problème de la distribution, nous voulons faire une simulation pour déterminer le chemin ou l'itinéraire de la distribution des fruits et légumes qui vise à trouver quels canaux de distribution fournissent l'itinéraire le plus court afin de fournir des coûts minimaux.

Samana, et al (2015) ont mené des recherches sur des simulations avec cette méthode de recuit. Le problème simulé est le Traveling Salesman Problem (TSP), pour vérifier la disponibilité des pièces de



rechange [1]. Dans le cas du TSP, il s'agit de rechercher le chemin le plus court et d'obtenir le temps et le coût optimaux en simulant la plus courte distance [2][3]. Recherche sur l'optimisation de la distribution de marchandises en fonction de la route et de la capacité en utilisant la méthode du recuit simulé [4][5]. Dans cette recherche, une application d'optimisation est faite pour distribuer des marchandises en considérant la route de déplacement. Le recuit simulé est également utilisé pour l'ordonnancement de la production [6] et l'ordonnancement des cours [7]. Ainsi, la méthode du recuit simulé peut être utilisée dans différents cas. Ainsi, nous voulons mettre en œuvre le système dans le processus de distribution de fruits dans la ville de Bogor afin de minimiser les canaux de distribution.





## 2. Théorie

### 2.1 Le problème du voyageur de commerce

TSP est un problème d'optimisation combinatoire. Les problèmes mathématiques de TSP ont été soulevés en 1800 par le mathématicien irlandais William Rowan Hamilton et le mathématicien anglais Thomas Penyngton [8]. Le TSP est une collection de villes et de coûts de voyage (ou distances) donnés entre chaque paire de villes utilisée pour trouver la meilleure façon de visiter toutes les villes et de retourner au point de départ dans un effort pour minimiser le coût ou la distance du voyage [9]. Le problème du voyageur de commerce (TSP) consiste à déterminer le chemin le plus court qui peut passer par tous les points et une seule fois. Par exemple, pour 3 points, il y a deux chemins, à savoir A-B-C-A ou A-C-B-A qui sont les directions opposées de A-B-C-A ou pour 4 points, il y a 3 chemins à savoir A-B-C-D-A, A-D-B-C-A ou A-D-C-B-A et 3 directions opposées.

Pour un certain nombre de voies avec des points très grands, cela prend beaucoup de temps à réaliser. En cas de test manuel, il n'est pas possible d'essayer tous les chemins possibles, mais de choisir la meilleure solution possible à partir du chemin existant. Pour résoudre ce cas, une méthode de recuit simulé est donc nécessaire [10]. Dans le problème du voyageur de commerce (TSP), le but atteint est l'itinéraire avec la plus courte distance, et la limite est que toutes les villes doivent être traversées et que chaque ville n'est parcourue qu'une seule fois. Un état moyen est défini comme un chemin possible et s'écrit avec  $S = [1, N]$

Ainsi, l'état dans une TSP est un ensemble de nombres de points de 1 à N et il n'y a pas de nombres égaux.

### 2.2 Recuit simulé

La résolution d'un problème d'optimisation à l'aide du recuit simulé s'inspire d'un processus physique, à savoir le refroidissement d'un métal appelé recuit. Le processus de recuit peut être défini comme une lente chute de température d'un objet qui a été préalablement chauffé jusqu'à un état où l'objet atteint un point de congélation ou, en d'autres termes, gèle le point de congélation. Le principe de fonctionnement est le suivant : à haute température, les particules solides et liquides ont un niveau d'énergie élevé, ce qui leur permet de se déplacer relativement facilement contre d'autres particules. Lorsque la température est abaissée, les particules s'organisent pour trouver un arrangement stable avec un niveau d'énergie minimal. Voici un tableau d'analogies entre le processus de recuit dans le refroidissement des métaux et le recuit dans les problèmes d'optimisation.

L'algorithme de recuit simulé a été introduit pour la première fois par Metropolis At Al. en 1959. L'algorithme de recuit simulé est un algorithme qui se déplace parfois vers des solutions avec des coûts élevés ou des solutions qui ne sont pas meilleures dans l'espoir que ce mouvement puisse exclure l'état du point minimum local [11]. La capacité d'accepter une mauvaise solution à un moment donné est ce qui distingue l'algorithme de recuit des autres algorithmes. L'acceptation de la solution à la situation est basée sur une méthode probabiliste développée par Kirkpatrick At. Al en 1983, à savoir l'acceptation d'une nouvelle solution basée sur une certaine probabilité pour trouver la valeur minimale globale d'une fonction qui a une valeur locale minimale.

Le recuit simulé sur la TSP est utilisé pour tracer et rechercher toutes les routes possibles, puis obtenir la route la plus courte. Le modèle de recuit simulé pour résoudre la TSP est un modèle d'état construit pour exprimer les routes possibles et les définitions de l'énergie exprimée par la distance totale parcourue [12].

Dans le recuit simulé, le processus de réduction de la température doit être pris en compte. Lors de l'itération initiale, la température doit être élevée pour que le processus de recherche aléatoire ait une large portée. Au fur et à mesure que l'itération augmente, la température continue de baisser mais peut ne pas atteindre zéro. Ainsi, avec le recuit simulé, on obtient des fonctions objectives de plus en plus petites ou décroissantes. Dans le recuit simulé, la "température" est un facteur qui fonctionne comme un facteur de contrôle pour l'acceptation ou le rejet du modèle. Lorsque la valeur de la température est élevée, les chances que le modèle soit accepté sont plus grandes, mais lorsque la température diminue ou a peu de valeur, de nombreux modèles sont rejetés. La température est la formation d'une substance utilisée comme fonction de contrôle dans le recuit simulé.

La simulation de recuit utilise le principe de la distribution de Boltzman. La simulation de recuit ne peut être acceptée que si la possibilité est

$$P = e^{-\frac{\Delta E}{kT}}$$





Où  $\Delta E$  est la différence entre l'énergie initiale et l'énergie finale,  $K$  est les constantes de Boltzman (0 et 1) et  $T$  est la température.

A partir de la fonction ci-dessus, si à  $\Delta E$  l'énergie finale ( $E_1$ ) est supérieure à l'énergie initiale ( $E_2$ ) alors  $E_1$  est reçu comme une nouvelle solution ponctuelle. Dans l'algorithme SA, il y a des étapes spéciales pour sortir de l'optimum.



solution locale. Cette étape consiste à recevoir un point d'énergie initial ( $E_1$ ) avec la probabilité  $e^{-\frac{\Delta E}{KT}}$  même si la valeur de la fonction à ce point n'est pas meilleure que celle du point précédent. Ceci est fait avec l'espoir que dans l'étape suivante, un point avec une meilleure valeur de la fonction sera atteint [7].

Certains éléments doivent être pris en compte dans la mise en œuvre du recuit simulé, à savoir :

1. État : défini comme une combinaison de la valeur du règlement possible de l'itinéraire emprunté pour passer par toutes les villes jusqu'au retour à la ville d'origine à condition que chaque ville soit traversée une fois. L'état est défini par  $S = (x_1, x_2, x_3, \dots, x_n)$ .
2. Énergie : définie comme la quantité de la fonction objective minimale d'une combinaison d'états. Dans le cas de TSP, l'énergie est définie comme la distance qui doit être parcourue sur un chemin qui est exprimé comme une séquence de numéros de ville à passer. L'énergie peut être exprimée par l'équation :

$$E = \sum_{i=1}^n d$$

Où  $E$  est l'énergie et  $d$  est la distance des villes à  $s(i)$  et  $s(i+1)$ . Alors que la distance ( $d$ ) est exprimée par :

$$d = \sqrt{(s_x(i) - s_x(i+1))^2 + (s_y(i) - s_y(i+1))^2}$$

3. Température : définie comme une valeur de contrôle qui fait monter ou descendre un état aléatoire. Comme l'analogie avec les événements thermiques, les ions se déplacent librement à haute température, et leurs mouvements deviennent plus limités lorsque la température baisse.
4. Mise à jour de l'état du processus : dans ce processus, l'état sera reçu simulé par ce recuit avec probabilité :

$$P = e^{-\frac{\Delta E}{KT}}$$

L'algorithme de recuit simulé (SA) est généralement utilisé pour résoudre tous les problèmes d'optimisation :

- a) Générer l'état initial  $S_0$  obtenu en générant des nombres aléatoires sur l'ordinateur et il ne doit pas y avoir de nombres jumeaux.
- b) Calculer l'énergie  $E_0$  au début de  $S_0$
- c) La mise à jour de l'état  $S$  avec les règles de mise à jour selon le problème devient  $S_i$
- d) Calculer l'énergie  $E_i$
- e) Générer des nombres aléatoires  $p = [0,1]$
- f) Si  $p < \exp(-\Delta E / T)$ , l'état est accepté, sinon, l'état est rejeté. Avec l'énergie  $E$  pour le problème TSP est défini comme suit :

$$E = \sum_{i=1}^{M-1} |x(s_{i+1}) - x(s_i)| + |x(s_1) - x(s_M)|$$

où

$E$  : La fonction d'énergie après itération  
 $x(s_i)$  : Position du point  $i$   
 $|x(s_i) - x(s_j)|$  : Distance entre le point  $i$   
 et le point  $j$   $M$  : Nombre de points visités.

- g) Abaisser  $T$  (température) avec la fonction de planification du refroidissement. Pour les cas de PST, la fonction de planification du refroidissement utilisée

$$T_i = T_0 \left( \frac{T_n}{T_0} \right)^{\frac{i}{N}}$$

où

$T_i$  : Température du programme de  
 refroidissement de  $i$   $T_0$  : Température initiale  
 $T_n$  : Température du programme  
 de refroidissement  $N$  : Nombre





d'itérations

i

: i-itération





h) Répétez la troisième étape jusqu'à ce qu'elle atteigne les critères d'arrêt.

L'algorithme du recuit simulé est basé sur l'algorithme de Metropolis qui est utilisé pour obtenir une bonne configuration. L'avantage du recuit simulé par rapport aux autres méthodes est sa capacité à éviter les pièges des optima locaux. L'algorithme est un algorithme de recherche aléatoire, mais il n'accepte pas seulement une valeur objective qui descend toujours, mais reçoit parfois une valeur objective qui monte.

### 3. Méthode de recherche

Dans cette section, un algorithme de recuit simulé (SA) pour le problème TSP sera appliqué à 22 emplacements de clients de fruits et légumes dans la ville de Bogor. Les données utilisées sont sous forme de coordonnées avec les données suivantes :

TABLEAU 1  
DONNÉES DE COORDONNÉES DE L'EMPLACEMENT

Numéro d'emplacem	Coordonnée (x)	Coordonnée (y)
1	3,002556	45,846117
2	0,644905	44,896839
3	1,380989	43,470961
4	1,376579	43,662010
5	5,337151	43,327276
6	7,265252	43,745404
7	1,650154	47,385427
8	1,430427	48,197310
9	2,414787	48,953260
10	3,090447	50,612962
11	5,013054	47,370547
12	4,793327	44,990153
13	2,447746	44,966838
14	1,750115	47,980822
15	4,134148	49,323421
16	7,506950	48,580332
17	1,233757	45,865246
18	4,047255	48,370925
19	0,103163	49,532415
20	1,495348	49,667704
21	4,494615	48,447500
22	0,457140	46,373545

Sur la base de l'algorithme SA, les étapes de la méthode SA peuvent être présentées comme suit :

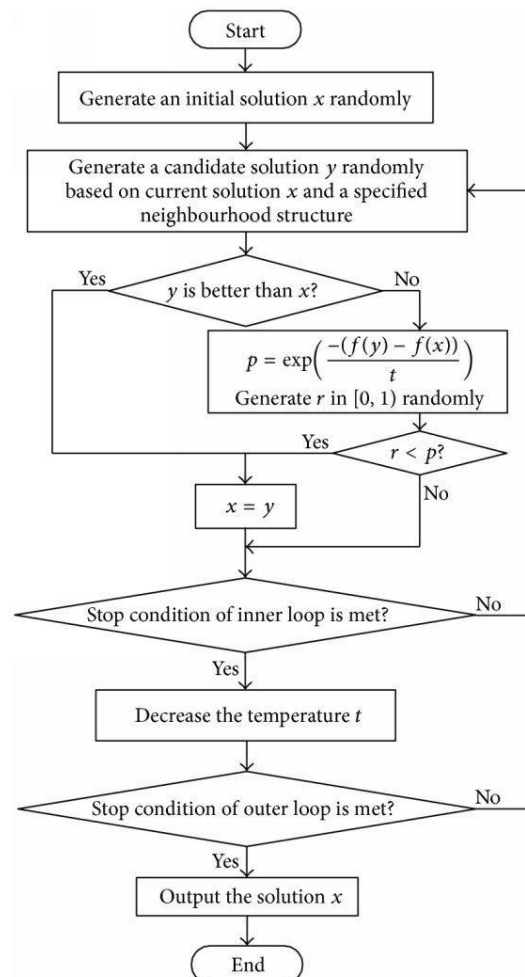


Figure 2. Organigramme de la méthode du recuit simulé [10]

Étapes de l'algorithme de recuit simulé avec un cas de 10 emplacements :

- Faites revivre l'état initial S0 obtenu en générant des nombres aléatoires sur l'ordinateur et il ne doit pas y avoir de nombres multiples.  
Appliqué à l'exemple avec 10 points de coordonnées dans le tableau 2 numéros 1 à 10 Obtenu l'état initial le long de 28.459728603126003 Km.
- Calculer l'énergie E0 au début de S0  
La distance qui a été parcourue sur un chemin qui indique l'ordre du numéro de la ville traversée.  
On a obtenu la valeur énergétique de : 25.151696358872837 Km
- Mettre à jour l'état S avec les règles de mise à jour en fonction du problème en cours Si  
Ainsi, S1 a été redéterminé comme = 25.151696358872837 Km, ceci parce que la nouvelle solution peut être acceptée (le nouvel état est meilleur que l'état précédent).
- Calculer l'énergie E1  
Le calcul de l'énergie avec l'état suivant produit E1 de 26.625758887223228 Km, donc la solution est rejetée car le nouvel état n'est pas meilleur que l'état précédent, alors on cherchera de nouvelles solutions aux 5ème et 6ème étapes.
- Générer des nombres aléatoires  $p = [0,1]$   
Parce que les résultats de la génération des nombres aléatoires  $(R) < p$ , alors la solution est acceptée qui est 26.625758887223228 Km





- f) Calculer l'énergie E2  
Le calcul de l'énergie avec l'état suivant donne E2 de 25.151696358872833 Km, la solution est acceptée.
- g) Abaissez la T (température) avec la fonction de planification du refroidissement.  
 $T1 = T0 - 0,0001$  ou  $0,5 - 0,0001 = 0,4999$
- h) Répétez la troisième étape jusqu'à ce qu'elle atteigne les critères d'arrêt.  
Dans ce cas, l'état optimal est obtenu à la 200ème itération avec les résultats suivants :  
24,034263695271754 Km

#### 4. Résultat

L'application du recuit simulé dans le cas de TSP, sera simulée aux endroits qui doivent être visités, en un voyage affichant la visualisation à chaque itération jusqu'à ce qu'il atteigne sa position optimale, dans ce cas la route circonférentielle la plus courte. Dans chaque simulation, les mêmes paramètres d'entrée sont donnés, tels que : Programme de refroidissement ( $T_n = 0,0001$ ), et itération ( $N = 1000$ ).

Voici les résultats pour les 4 points d'implantation avec les résultats suivants : état initial de la simulation, état final de la simulation et réduction d'énergie.

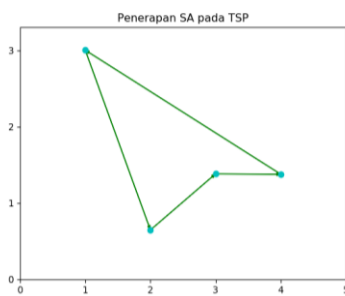


Figure 2. Iteration de l'état initial

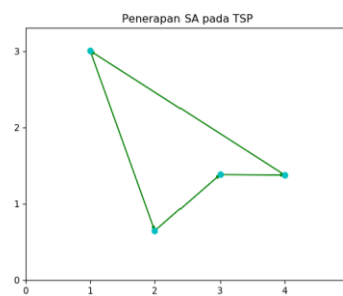


Figure 3. Iteration de l'état final

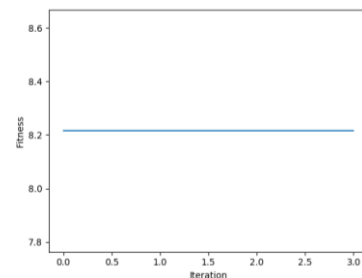


Figure 4. Réduction de l'énergie

Voici les résultats pour les 10 points d'implantation avec les résultats suivants : état initial de la simulation, état final de la simulation et réduction d'énergie.

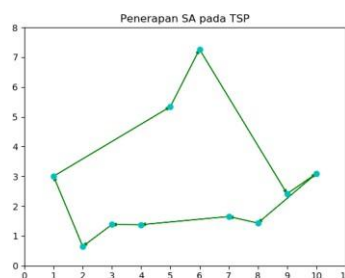


Figure 5. Iteration de l'état initial

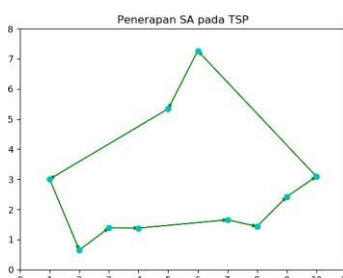


Figure 6. Iteration de l'état final

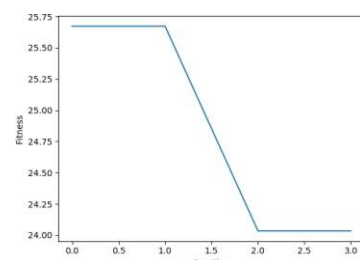


Figure 7. Réduction de l'énergie

Distance obtenue à l'état final : 24.034263695271754 Km et diminution de la distance après optimisation : 6.38%.

Voici les résultats pour les 22 points d'implantation avec les résultats suivants : état initial de la simulation, état final de la simulation et réduction d'énergie.

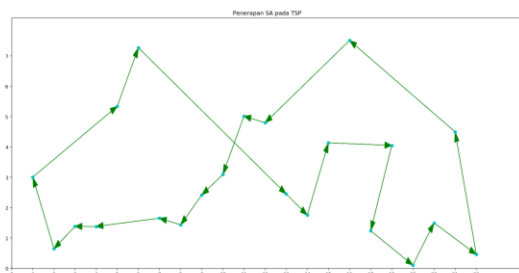


Figure 8. Itération de l'état initial

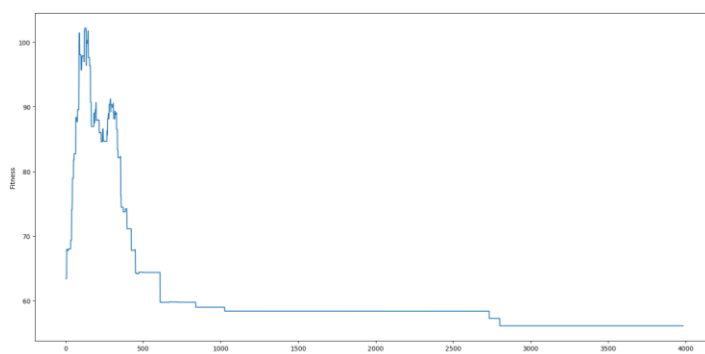
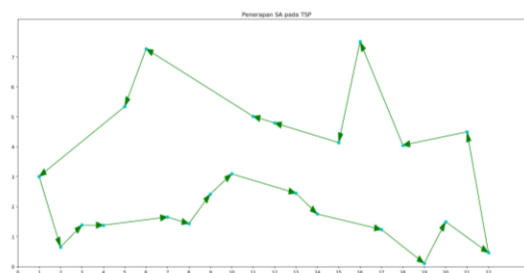


Figure 10. Réduction de l'énergie

Distance obtenue dans l'état final : 56.134938628085166 Km et diminution de la distance après optimisation : 7.72%.

## 5. Conclusion

Pour les cas de 4 et 10 points de localisation, il suffit de faire 1000 itérations pour obtenir les résultats optimaux, tandis que pour le cas de 22 points de localisation, il suffit de faire 100000 nouvelles itérations pour obtenir les résultats optimaux. On peut voir sur le graphique de la réduction d'énergie pour chaque point de localisation où pour 4 points de localisation il est suffisant de faire avec 1 itération. Alors que pour 10 emplacements, 200 itérations sont nécessaires. Dans le cas de 22 points, 1000 itérations n'ont pas abouti à un état final optimal, on peut le voir sur l'affichage final de la simulation, où il y a encore beaucoup de routes qui se croisent. En outre, le graphique de diminution de l'énergie ne semble pas stationnaire. L'itération nécessaire pour atteindre les conditions optimales pour le cas de 22 points est de 2800 itérations. Ceci est indiqué par la tendance du graphique après une itération supérieure à 2800 à être stationnaire (plat). Le nombre d'itérations qui produisent des conditions/solutions optimales dépend du nombre de points de localisation simulés. Plus le nombre de points de localisation simulés est élevé, plus le nombre d'itérations nécessaires est important. Les résultats de la simulation à partir de 4, 10 et 22 points de localisation montrent que l'itinéraire optimal est l'itinéraire le plus éloigné et qu'il n'y a pas de croisement d'itinéraires.

## 6. Références

- [1] E. Samana, B. Prihandono, et E. N. Intisari, "Aplikasi Simulated Annealing Untuk Menyelesaikan Travelling Salesman Problem", *Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster)* Volume 03, n° 1, pp. 25 - 32, 2015.
- [2] Citra Nur Alpianty et J. Dharma Lesmono, "Penerapan Algoritma Simulated Annealing Untuk Menyelesaikan Asymmetric Travelling Salesman Problem", in *Seminar Nasional Matematika Vol.13*, 2018, pp. 63-69.
- [3] A. Meitrirova, F. M. Puspita, and Yahdin, "Implementasi Travelling Salesman Problem (Tsp) Dengan Algoritma Genetika Dan Simulated Annealing Pada Pendistribusian Pupuk Pt. Austindo Nusantara Jaya,"





Sriwijaya

# JurnalMantik

Volume 3 Numéro 4, Février 2020, pp. 611-618

E-ISSN 2685-4236

<https://iocscience.org/ejournal/index.php/mantik/index>





- Université, 2017.
- [4] R. Hidayati, I. Guntoro, et S. Junianti, "Penggunaan Metode Simulated Annealing Untuk Penyelesaian Travelling Salesman Problem," 2019.
  - [5] Jose Rizal, "Optimasi Pada Traveling Salesman Problem (TSP) dengan Pendekatan Simulasi Annealing," *J. Gradien*, vol. 3, no. 2, pp. 286-290, 2007.
  - [6] M. Hamdani Azmi et C. Farela Mada Tantrika, "Penjadwalan Produksi Rokok Untuk Meminimalkan Maximum Tardiness Menggunakan Algoritma Simulated Annealing," *J. Rekayasa Dan Manaj. Sist. Ind.*, vol. 3, n° 2, p. 353-362, 2015.
  - [7] Dany Purno Yuwono, "Penerapan Simulated Annealing Untuk Penjadwalan Matakuliah Di Jurusan Teknik Elektro Universitas Negeri Malang", Universitas Negeri Malang, 2014.
  - [8] S. Vats et N. Bakshi, "Review Paper on Genetic Algorithm and Simulated Annealing for solving Travelling Salesman Problem", *Int. J. Innov. Res. Technol.* vol. 5, no. 3, pp. 272-275, 2018.
  - [9] Fatmawati, Bayu Prihandono, et Evi Noviani, "Penyelesaian Travelling Salesman Problem Dengan Metode Tabu Search.", *Bul. Mat. Stat. dan Ter.* vol. 4, no 1, p. 17-24, 2015.
  - [10] L. Wang, M. L. R. Cai, et Y. Zhong, "Enhanced List-Based Simulated Annealing Algorithm for Large-Scale Traveling Salesman Problem," *IEEE Xplore Digit. Libr.*, pp. 144366-144380, 2019.
  - [11] ChangYing Wang, Min Lin, YiWen Zhong et Hui Zhang, "Solving travelling salesman problem using multiagent simulated annealing algorithm with instance-based sampling", *Int. J. Comput. Sci. Math.* vol. 6, no. 4, 2015.
  - [12] Hüsamettin Bayram et Ramazan Şahin, "A New Simulated Annealing Approach for Travelling Salesman Problem," *Math. Comput. Appl.* vol. 18, no. 3, pp. 313-322, 2013.

