Chapter 2

A New Approach towards ACO through Punishment Mechanism

Chapter 2

A New Approach towards ACO through Punishment Mechanism

2.1 Punished Elitist Ant System and Punished Rank Ant System

2.1.1 Introduction

As discussed in the chapter 1, a fundamental principle in ACO is the ant trial formation in which indirect interaction of the ants takes place through the media of deposited chemical pheromones. These communicative process enables ant colonies to make adaptive choices based on local information. This stigmergy concept, in general allows for a complex collective cooperative behavior of simpler agents and inspired the genesis of quite a few number of algorithms and applications.

However, efficient flow of ants is not only concerned with shortest paths, but also coping with the characteristics of the ants. An inefficient regulation of ants flow will usually lead to a bottle neck situation. This is one of the most challenging tasks in ACO. As the efficient distribution of limited resources by decentralized individual decisions is still an open problem in many network related problems, investigation of ant's behaviors needs to be done cautiously and taking necessary actions on the performance of ants will be the right move towards the betterment of the algorithm.

Before we propose a new version of ACO with certain actions taken on the 'poorly performed' ants, the brief overview of the generic pheromone model in ACO is necessary, which was broadly explained in the section 1.4.1. An isolated ant moves randomly, but an ant encountering a multiple previously accumulated pheromone trial paths takes decision to traverse on one of them and thereby, reinforce it with more pheromones. The repetitions of above mechanism represent the auto-catalytic behavior of real ant colony. Further, ACO algorithms use a colony of ants which are allowed to search and reinforce the pathways in order to find the optimal ones. After initialization of the pheromone trails, ants construct feasible solutions starting from the random nodes and then updates the pheromone trials. At each step ants compute a set of feasible moves and select the best one to carry out the rest of the tour. The transition probability is based on the heuristic information and pheromone trial. In the beginning, the initial pheromone level will be set to a small positive constant value and then ants update these values after completing the construction phase. Different ACO algorithms adopt different criteria to update the pheromone level. A new concept of punishing the 'non-performing ants' is introduced to regulate the pheromone level at the updation stage and this mechanism is termed as punishment mechanism.

2.1.2 Punishment Mechanism

The idea of punishment mechanism is conceived from a special class of functions appearing in literature on constrained combinatorial optimization problems, called as penalty functions. Of course, penalty functions have been a part of the scientific articles related to the above mentioned areas for decades (Schwefel, 1995 and Coello, 2002). Boardly, penalty functions are used to restrict the search to feasible solutions and to give a scheme that will drive the population towards the optimum (Michealewicz, 1995). While incorporating distance together with the length of the search, into the penalty functions has been generally effective. In this respect, they are adaptive to the ongoing success of the search and cannot guide the search towards any particularly attractive regions or away from unattractive regions based on what has already been observed. Thus, these penalty based methods in Evolutionary Algorithms are quite popular and their strengths, weaknesses are throughly discussed in the paper due to Yeniay (2005).

The idea of penalization can be applied to ACO also. The implication of punishment mechanism on ACO is that, it draws a boundary across the promising regions in a search space and within this boundary, ants are forced to search for the optimal solution. In this chapter, we introduce and discuss the incorporation of punishment mechanism in ACO. This approach involves some new equations, which are carefully chosen such that, evaluation process is not long, otherwise a lot of computational function assessments may be required, thereby making algorithm less practical. A variety of constraint handling methods have been suggested by the researchers. Each method has its own merits and demerits. The main problem is to define the punishment process mathematically. Consequently, researchers have to experiment with different formula of punishment mechanism for different problems. The generic pseudo-code for the Punished Ant Colony Optimization is given by Algorithm 5:

Algorithm 5 Punished Ant Colony Optimization

```
Initialize the pheromone values.
while termination conditions not met do
  START ScheduleActivities
  ConstructAntsSolutions
  UpdatePheromone
  UpdateEliteAntPheromone
  UpdateNonEliteAntPheromone
  DeamonActions
  END ScheduleActivities
end while
```

The punished ACO pseudo-code has additional two new procedures namely *UpdateEliteAntPheromone* and *UpdateNonEliteAntPheromone* in the conventional ACO. The *UpdateEliteAntPheromone* procedure reinforces the path traveled by the elite ants and *UpdateNonEliteAntPheromone* procedure removes specified quantity of pheromone trial on a non-elite paths. We discuss the incorporation of punishment mechanism to some variants of ACO and give detailed analysis in the coming sections.

2.1.3 Punishment Mechanism in Elitist Ant System (EAS) and Rank Ant System (RAS)

The concept of elitism was introduced in AS by Dorigo et al. (1996) and it was further extended to rank the ants by Bullnheimer (1999), where pre-specified selected number of best (elite) paths of the iteration will be reinforced for the second time. The basic purpose of additional reinforcement is to ensure that search for optimal solution remain in the promising area of the search space. However, it is possible to restrict further by incorporating the punishment mechanism. The punishment mechanism suggests the removal of certain amount of pheromone trial on a nonelite paths. Thus, pheromone removal process decreases the probability of selecting the non-elite paths and ants are forced to search in the neighborhood of the promising solutions.

The punishment mechanism has been incorporated into two versions of ant systems namely, Elitist ant system and Rank ant system. Here onwards these new algorithms are called Punished Elitist Ant System (PEAS) and Punished Rank Ant System (PRAS) respectively. The basic purpose to introduce the punishment mechanism is to favor exploitation over exploration by restricting the search in promising area of the search space. The quantity of pheromone to be removed will be specified by the punishment specification in the algorithm. The punishment specification for PEAS specifies to decrease the quantity of pheromone trial proportional to the quality of solution found on a non-elite paths and in case of PRAS, all the non elite paths are weighted according to their performances and then proportionately decreased. The generic punishment feature is given by the equation:

$$\tau_{ij} = \tau_{ij} - \Delta\tau_{ij}^{*} \tag{2.1}$$

where $\Delta\tau_{ij}^{*}$ is

$$\Delta\tau_{ij}^{*} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k}$$

The $\Delta\tau_{ij}^*$ represents the amount of pheromone trial decrease on a path ij due to l number of non-elite ants.

Punishment Specification for PEAS

The punishment specification for PEAS directs to decrease the quantity of pheromone trial proportional to the quality of solution found on a nonelite paths.
$\Delta\tau_{ij}^k$ for PEAS is given by the equation:

$$\Delta\tau_{ij}^k = \begin{cases} l\cdot Q^*/L_k & \text{if (i, j)} \in k^{th} \text{ ant's non performing tour} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where Q^* is the algorithmic constant and L_k represents the tour length of the k^{th} ant. If e represents the number of elite ants, then l represents the number of non-elite ants given by the expression $l = m - e$.

Punishment Specification for PRAS

The punishment specification for PRAS directs that the quality of solution found on a non elite paths must be ranked according to their performances and then proportionately decreased.
$\Delta\tau_{ij}^k$ for PRAS is given by the equation:

$$\Delta\tau_{ij}^k = \begin{cases} Q^*\cdot (l-k)/L_k & \text{if (i, j)} \in k^{th} \text{ ant's non performing tour} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

It can be observed from equation (2.1) that amount of pheromone removed from a path will be proportional to the quality of solution found. Thus, a better infeasible solution will receive less punishment than the inferior one.

2.1.4 Experimental Results and Performance Analysis

In order to demonstrate the superiority of the proposed algorithms, a comparative analysis was performed with following existing ant algorithms: AS, EA, RA and variants of MMAS - MMAS with Global best updation (MMAS+GB) , MMAS with Global best updation and Pheromone trial smoothing (MMAS+GB+PTS), MMAS with Iteration best updation (MMAS+IB) and MMAS with Iteration best updation and Pheromone trial smoothing (MMAS+IB+PTS).

Parameter Settings

Inorder to assess the performance of algorithms, parameters α, β were varied from 1 to 5 and ρ was varied from 0.7 to 1.0. The algorithms were executed 10 times independently by considering some of the datasets available in the TSPLIB (see Table 2.1). The maximum number of iterations was set to $1,00,000$. The dataset considered for experimentation purpose are as follows:

Table 2.1: Datasets for ACO algorithms.

Datsets
Bays29
Att48
Eil51
St70
Eil76
Kroa100
Kroa200
Lin318

Computational Results and Comparative Analysis

Table 2.2 shows the comparative results of algorithms with that of existing ones for best solution, average solution and percentage of deviation from the optimal solution. The average solution was computed using the best solutions of last 50 iterations. The proposed algorithms were able to find the better solutions and improved averages for most of the datasets. The observed best solution for PEAS variant has a deviation of 0.02% for eil51 and lin318 datasets and in case of PRAS, it is 0% for kroa100, kroa200 and lin318 datasets. Table 2.2 shows that PEAS algorithm provides better solution for smaller problem dimension. The maximum deviation in best solution observed is 0.27% in eil76 and kroa100, which is comparatively lesser than the other algorithms. Similarly, the PRAS algorithm gives slightly better solution than the PEAS algorithm for higher dimension problem and the maximum deviation observed is 0.56% for st70 dataset. Another interesting observation is that, average solutions of PEAS and PRAS have smaller deviation from the optimal solution compared to other existing algorithms. The maximum deviation observed in case of PEAS is 0.79% and in case of PRAS is 0.81%. Hence, it can be concluded that, these algorithms succeed in restricting the search in promising region of search space.

Table 2.2: Performance comparision of PEAS and PRAS for various datasets with other variants of ant algorithm.

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
bays29	AS	2065.6 (2.25%)	2078.3 (2.88%)
	MMAS+GB	2045.2 (1.24%)	2053.3 (1.64%)
	MMAS+GB+PTS	2036.4 (0.81%)	2043.7 (1.17%)
	MMAS+IB	2032.1 (0.59%)	2038.1 (0.89%)
	MMAS+IB+PTS	2022.1 (0.1%)	2025.3 (0.26%)
	EA	2040.6 (1.01%)	2052.4 (1.6%)
	RA	2030.2 (0.5%)	2037.8 (0.88%)
	PEAS	**2021.7 (0.08%)**	**2024.3 (0.21%)**
	PRAS	2025.7 (0.28%)	2028.9 (0.44%)
att48	AS	10880.6 (2.37%)	10895.3 (2.51%)
	MMAS+GB	10690.8 (0.59%)	10704.5 (0.71%)
	MMAS+GB+PTS	10666.4 (0.36%)	10680.7 (0.49%)

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	MMAS+IB	10635.5 (0.07%)	10645.8 (0.16%)
	MMAS+IB+PTS	10634.4 (0.06%)	10640.8 (0.12%)
	EA	10810.6 (1.71%)	10825.6 (1.85%)
	RA	10728.4 (0.94%)	10740.6 (1.05%)
	PEAS	10640.7 (0.11%)	10654.8 (0.25%)
	PRAS	10680.1 (0.49%)	10694.4 (0.62%)
eil51	AS	432.6 (1.54%)	437.3 (2.65%)
	MMAS+GB	430.4 (1.03%)	434.1 (1.89%)
	MMAS+GB+PTS	428.5 (0.58%)	429.2 (0.75%)
	MMAS+IB	426.5 (0.01%)	427.8 (0.43%)
	MMAS+IB+PTS	426.2 (0.04%)	427.8 (0.43%)
	EA	427.1 (0.25%)	428.1 (0.49%)
	RA	430.2 (0.98%)	434.5 (1.99%)
	PEAS	**426.1 (0.02%)**	**426.5 (0.11%)**
	PRAS	426.3 (0.07%)	**426.8 (0.18%)**
st70	AS	705.3 (4.48%)	711.6 (5.42%)
	MMAS+GB	695.6 (3.05%)	702.6 (4.08%)
	MMAS+GB+PTS	688.6 (2.01%)	693.6 (2.75%)
	MMAS+IB	676.9 (0.28%)	682.7 (1.14%)
	MMAS+IB+PTS	675.5(0.07%)	680.3(0.78%)
	EA	695.3 (3%)	700.8 (3.82%)
	RA	682.6 (1.12%)	688.8 (2.04%)
	PEAS	**675.5 (0.07%)**	**678.9 (0.57%)**
	PRAS	678.8 (0.56%)	680.5 (0.81%)
eil76	AS	580.3 (7.8%)	592.6 (10.14%)
	MMAS+GB	547.4 (1.74%)	552.7 (2.37%)
	MMAS+GB+PTS	544.6 (1.22%)	548.9 (2.02%)
	MMAS+IB	539.2 (0.22%)	542.6 (0.85%)

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	MMAS+IB+PTS	538.5 (0.09%)	539.9 (0.35%)
	EA	560.7 (4.21%)	565.3 (5.07%)
	RA	545.6 (1.41%)	551.4 (2.49%)
	PEAS	539.5 (0.27%)	541.8 (0.7%)
	PRAS	**538.1 (0.01%)**	**539.2 (0.22%)**
	AS	21913.5 (2.96%)	22471.4 (5.58%)
	MMAS+GB	21722.3 (2.06%)	21814.7 (2.50%)
	MMAS+GB+PTS	21350.3 (0.32%)	21417.1 (0.64%)
	MMAS+IB	21310.8 (0.13%)	21373.2 (0.43%)
Kroa100	MMAS+IB+PTS	21285.4 (0.01%)	21336.9 (0.26%)
	EA	21780.6 (2.34%)	21890.7 (2.86%)
	RA	21612.7 (1.55%)	21746.0 (2.18%)
	PEAS	21340.8 (0.27%)	21450.6 (0.79%)
	PRAS	**21283.3 (0%)**	**21390.8 (0.51%)**
	AS	31156.7 (6.09%)	31210.3 (6.27%)
	MMAS+GB	29546.8 (0.6%)	29575.3 (0.7%)
	MMAS+GB+PTS	29482.7 (0.39%)	29520.9 (0.52%)
	MMAS+IB	29421.3 (0.18%)	29445.8 (0.26%)
kroa200	MMAS+IB+PTS	29372.2 (0.01%)	29385.8 (0.06%)
	EA	29870.6 (1.71%)	29921.6 (1.88%)
	RA	29744.4 (1.28%)	29781.7 (1.4%)
	PEAS	**29385.8 (0.06%)**	29411.3 (0.14%)
	PRAS	**29370.6 (0%)**	**29382.4 (0.04%)**
	AS	42780.3 (1.78%)	43139.9 (2.64%)
	MMAS+GB	42593.8 (1.34%)	42647.1 (1.47%)
	MMAS+GB+PTS	42454.8 (1.01%)	42496.2 (1.11%)
	MMAS+IB	42280.9 (0.59%)	42289.2 (0.61%)
lin318	MMAS+IB+PTS	42035.7 (0.01%)	42055.8 (0.06%)

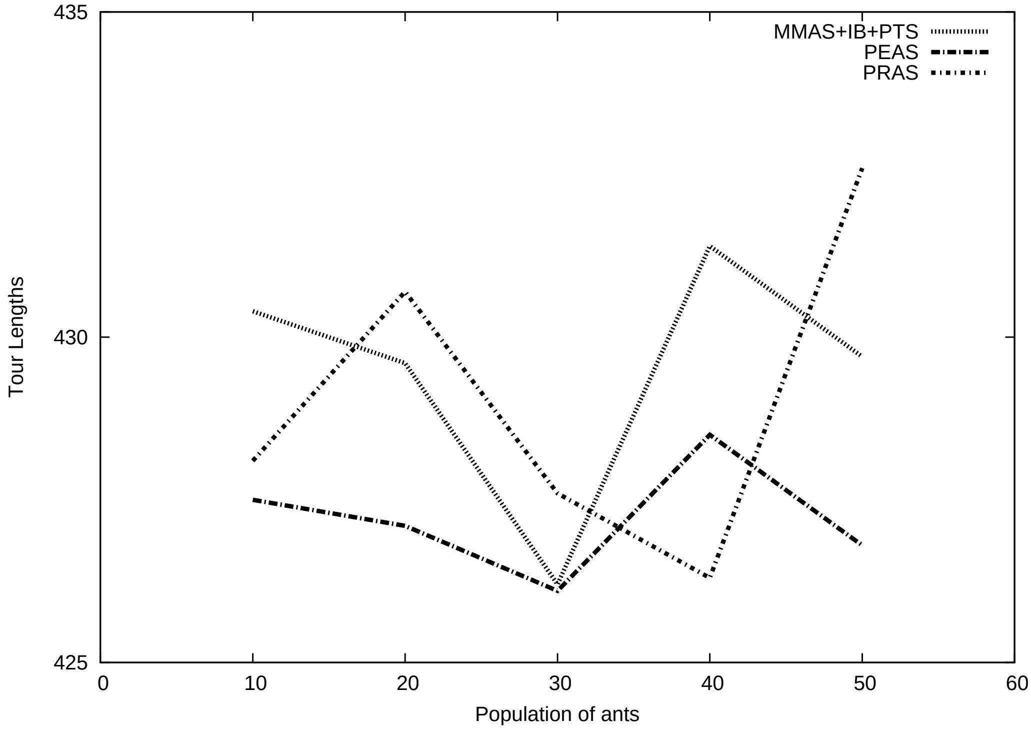
Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	EA	42670.6 (1.52%)	42711.9 (1.62%)
	RA	42467.1 (1.04%)	42491.8 (1.10%)
	PEAS	42038.6 (0.02%)	**42057.8 (0.06%)**
	PRAS	**42032.1 (0%)**	42065.4 (0.08%)

Parameter Sensitivity Analysis

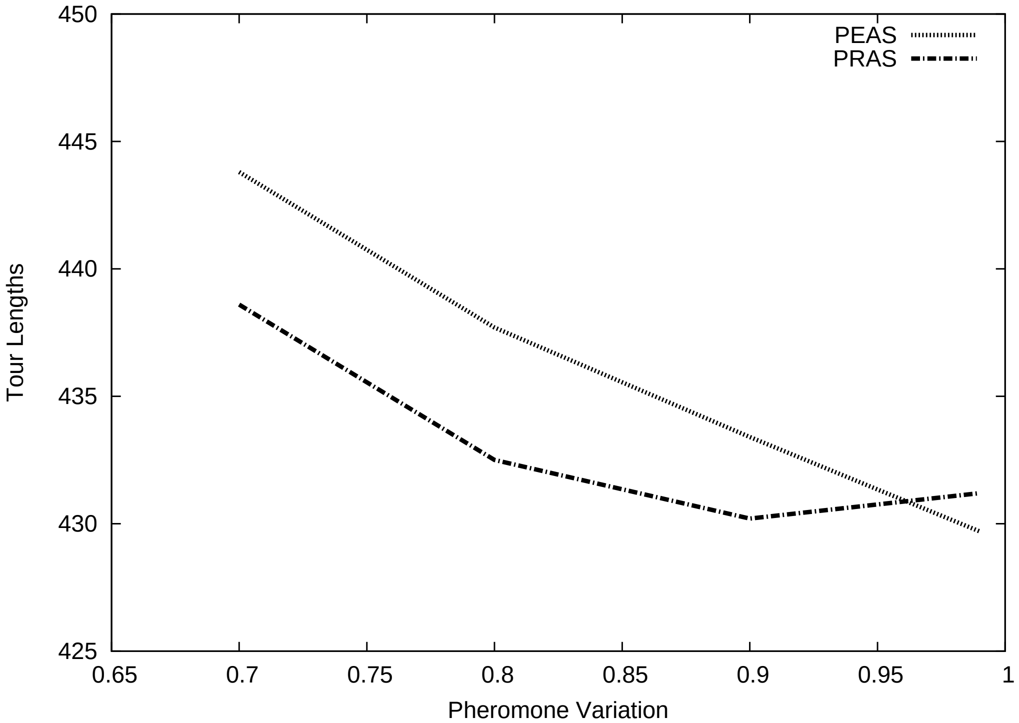
To assess the parameter sensitivity of the algorithms, a comparative analysis were performed with the best algorithm i.e., MMAS+IB+PTS available in the literature. Figure 2.1(a) shows the comparison of the algorithms with MMAS+IB +PTS for the smaller dimension eil51 dataset. It can be observed that PEAS algorithm exhibits less variation in best solution compared to other algorithms for varying number of ants and provide best result for m=30. Figure 2.1(b) shows the comparison graph of PEAS and PRAS variants for varying pheromone trial strength. The proposed variants are sensitive to the amount of pheromone trial and perform better for higher pheromone strength. The PEAS variant provide best result for ρ=0.99 and PRAS for ρ=0.9.

Similarly, Figure 2.2 shows the variation of best solution for varying number of ants for larger dataset Kroa100, where PRAS exhibits lesser variation compared to other ant algorithms and provide best result for m=25. It can be further observed that performance of PEAS is inferior to both PRAS and MMAS+IB+PTS variants.

Figure 2.3 illustrates the variation of best solution for varying number of punished ants. The number of ants m was set to 50. It can be observed that for PEAS algorithm, small number of punished ants provides better result than the PRAS algorithm. As the number of punished ants increases, quality of solution decreases thereby indicating the difficulty to find the solution in a small search space. The solution quality deteriorates much faster for PEAS than the PRAS variant.



(a) Variation in number of ants



(b) Variation in pheromone trial

Figure 2.1: Comparision of PEAS and PRAS algorithms for eil51 dataset.

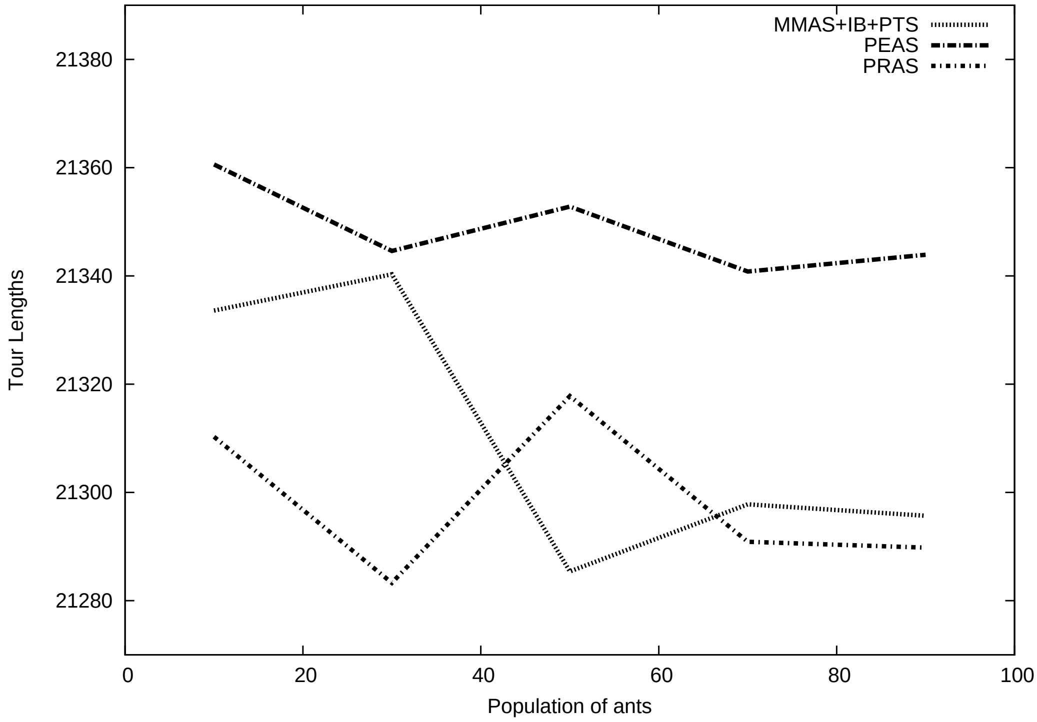


Figure 2.2: Comparision of PEAS and PRAS algorithms for variable number of ants for Kroa100 dataset.

Table 2.3: Parameter details for PEAS and PRAS

PEAS	$1 \leq \alpha \leq 2$	$3 \leq \beta \leq 4$	0.8 - 0.9
PRAS	$2 \leq \alpha \leq 4$	$1 \leq \beta \leq 3$	0.76 - 0.85

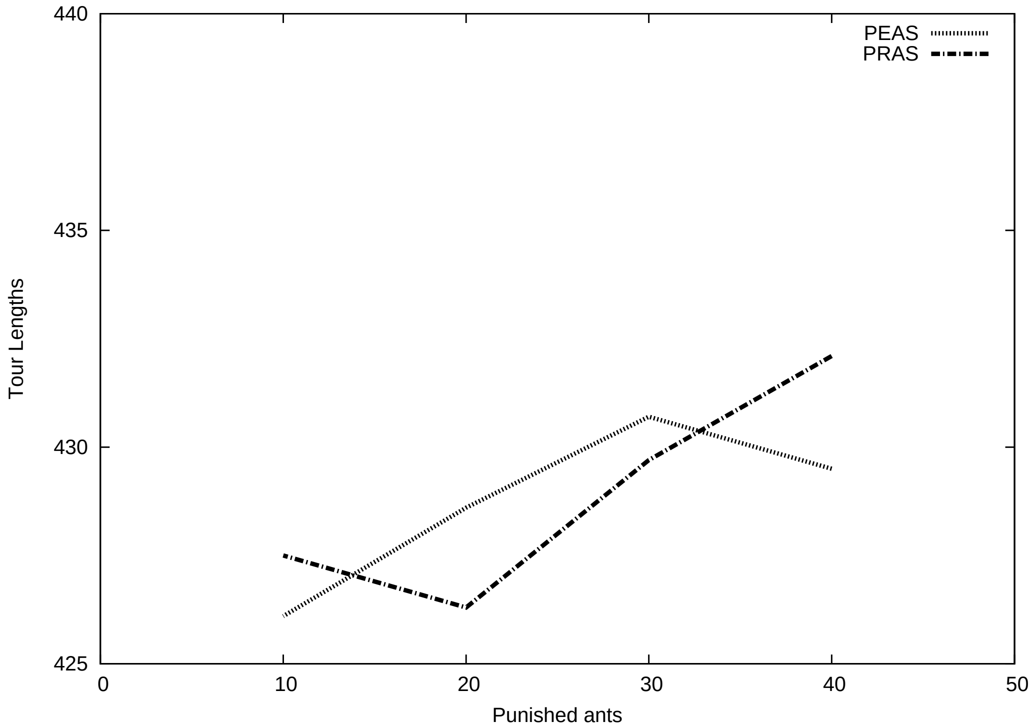


Figure 2.3: Comparison graph of PEAS and PRAS algorithms for variable number of punished ants.

Table 2.3 reports the observed parameter values for PEAS and PRAS algorithms, when the optimal solutions were obtained. The parameter values are specified in range bounds to indicate the most common observations for the datasets under consideration. For example, α varies from 1 to 2, β varies from 3 to 4 and ρ varies from 0.8 to 0.9 for most of the datasets used for PEAS algorithm. Similarly, α varies from 2 to 4, β varies from 1 to 3 and ρ varies from 0.76 to 0.85 for PRAS variant. It can be concluded that PEAS provides better result for lower α, higher β, ρ values and PRAS provides better result for higher α and wider range of values for β and ρ.

2.2 Performance Linked Elitist Ant System and Performance Linked Rank Ant System

2.2.1 Introduction

The previous section discussed about the incorporation of punishment mechanism in EAS and RAS algorithms. The search mechanism were static in nature and it is controlled by composition of elite ants and non-elite (punished) ants that are specified as parameter settings. However, it is possible to induce dynamicity in search space by using appropriate elite selection mechanism. The elitism is multi-factor dependent. As an operational feature of ACO, elitism provides a means for improvising path drift towards feasible solutions by ensuring that best elite ants are allowed to reinforce the paths. Some ants of high elitism may turn out to be more important to the final solution than others. It is important to maintain an adequate selection pressure of elite ants as demanded by the applications. Since, static elitism can increase the path selection pressure by preventing the loss of pheromones due to equal treatment on any elitistic ant. Thus, as an adjustment of degree of elitism, we can introduce a concept of elite selection mechanism in ACO.

2.2.2 Machine Learning

Machine learning(ML)(Ethem, 2005) is a discipline of computing field concerned with training the machines to perform certain tasks. A machine learns to perform the task by gaining knowledge and by remembering the past experiences. The acquired knowledge and experience will be used to make the decisions that are necessary to solve the task. The underlying basis for learning mechanism is the statistical data collected from the observation and the data that will evolve in the future. The collected data will be used to train the machine, so that it can take appropriate decision for solving the task and the evolved data is used to adjust the decision making ability of the machines in order to improve the accuracy and performance. The ultimate goal of the ML is to mimic the human intelligence in machines. Machine learning is an interdisciplinary field and borrowed the idea from other fields like statistics, pattern recognition, artificial intelligence, adaptive control theory, evolutionary models etc. A machine can be trained to solve some of the tasks like (Han and Kamber, 2004);

- **Classification** - The process in which machine will act as a classifier and assign the data to the groups/classes they belong or assign the label based on group for the previously unseen data.

- **Prediction** - The mechanism is similar to classification, where a machine predicts the class of the incoming data based on past experience and knowledge.

- **Rule Generation** - The process of generating the rules by looking into the relations that exist between the data.

- **Clustering** - The process of grouping the data based on the similarities that exist in the data.

There are many learning mechanisms in literature and some of them are:

1. **Supervised learning** - A machine is acquainted with the knowledge about the number of classes and characteristic features of each class. Initially, machine will be trained with few samples of data to perform the task. Classification (Tan et al., 2006) and prediction (Domingos and Pazzani, 1997) tasks fall under this category of learning.

2. **Unsupervised learning** - In this learning mechanism, machine have no knowledge about the number of classes and characteristic features about the classes. Infact, machine learns by performing task. Clustering (Tan et al., 2006) is an example that falls under this category of learning.

3. **Reinforcement learning** - A learning mechanism that specifies the action need to be taken for each observation and reward the action in the form of feedback that guides the learning process (Kaelbling et al., 1996).

The ML methods find lot of applications in real life and some of the applications like analyzing customer buying pattern in supermarket, face recognition, stock market prediction etc have been commercially deployed.

2.2.3 The Elitist Ants Selection Mechanism

In this section, we will discuss the incorporation of ML philosophy in elitist ants selection procedure. The elitist selection process can be treated as a two class classification problem, where we need to design the classifier that will automatically classify the ants depending upon their performances. The classifier will place the performing ants into one class and the non-performing ants into another class. It can be observed that, ants have to remember the class they belong to, in addition to tour length. The statistical tools will provide the *decision boundary* that separates the two classes. The following statistical tools have been used in the design of classifier:

- **Mid - Range Tour Selection (MRTS):** MRTS is the average of best tour length and worst tour length in a given iteration and is given by the equation:

$$MRTS = \frac{Best\ Tour\ Length + Worst\ Tour\ Length}{2} \tag{2.4}$$

- **Mean Tour Selection (MTS):** MTS is the mean of all the tour lengths in a given itertion and is given by the equation:

$$Mean = 1/n \sum_{i=1}^{n} TL_i \tag{2.5}$$

  where TL_i represents the tour length.

- **Median Tour Selection (MeTS):** MeTS is the median obtained by arranging all the tour lengths of a given iteration TL_i in the increasing order i.e., $TL_1 \leq TL_2 \leq \cdots \leq TL_n$ and $TL_{n/2}$ is selected as the median value.

The *MRTS, MTS* and *MeTS* values will act as boundary separating the performing and non performing ants.

2.2.4 Performance Linked Elitist Ants Selection Mechanism

The motivation for emergence of these types of algorithms comes from the following observation. The first observation is that the papers due to Dorigo et al. (1996) and Bullnheimer et al., (1999) do not discuss the criteria for selecting the elitist ants and all of the ants were considered as elitist. Infact, the program execution reveals that having fewer number of elitist ants will restrict the exploitation near the good solutions (or smaller regions) and may not contribute much to the final quality of solution. Similarly, if large number of ants were selected as elitist, then most of the paths will get additional reinforcement, leading to better exploration of search space. Although there will be improvement in quality of solution, but in due process, there will be reinforcement for some paths that may not contribute to the final solution. This necessitates the selection of optimum number of ants to strike the balance between exploration and exploitation.

The second observation is that ants gain information about the search space through exploration at the early stage of search process. It can be argued that size of the search space will be large at the initial stage and will get reduced at the later stage due to knowledge gained by the ants in exploration phase. The additional reinforcement of appropriate paths in exploration stage should help ants to look for better solution in exploitation stage. The search process should update comparatively larger number of elite paths during exploration phase and fewer number of paths upon transition to exploitation phase. It is interesting to observe that overall search region is going to be dynamic in nature and the dynamicity is introduced by selecting appropriate number of elite ants. The selection of elite ants is purely based on the performance of ants and is done using statistical functions.

The generic pseudo-code for performance linked Elitist ACO is given by Algorithm 6:

Algorithm 6 Performance Linked Elitist Ant Colony Optimization

```
Initialize the pheromone values.
while termination conditions not met do
    START ScheduleActivities
    ConstructAntsSolutions
    SelectEliteAnts
    UpdatePheromone
    UpdateEliteAntPheromone
    DeamonActions
    END ScheduleActivities
end while
```

The performance linked elitist ACO pseudo-code incorporates two new procedures namely *SelectEliteAnts* and *UpdateEliteAntPheromone* to the conventional ACO. The *SelectEliteAnts* procedure implements a classifier that is trained to classify the ants based on statistical functions and *UpdateEliteAntPheromone* procedure reinforce the selected elite paths choosen by *SelectEliteAnts* procedure.

2.2.5 Performance Linked EAS and Performance Linked RAS

This section discusses about a new approach towards ACO, which is an effective combination of existing ant algorithm and non-static elitism. A static elitism may lead to premature convergence to a sub-optimal solution. This is because a high selection pressure results in the population reaching equilibrium, but it sacrifices ants diversity. Thus, it is expected to identify the ants by the strength of elitism. This, strategy may gently push the simulated population towards the restoration of diversity among the elite ants.

The performance based elitist selection mechanism was applied to well known versions of ant algorithms namely EAS and RAS. The elite selection mechanism will place all the performing ants into one class and non-performing ants into another class based on statistical function. Suppose, if mean function is used as a classifier, then tour performances lesser than mean will be in performing class and the rest will be in non-performing class. The performing ants will get chance for additional reinforcement. Since the number of selected ants varies across the iteration, we will name the selected ants as *Influential Ants* (IA). The IA will provide the specification for reinforcement. These new algorithms here onwards are named as *Performance Linked Influential Elitist Ants System*(PLIEAS) and *Performance linked Influential Rank Ants System*(PLIRAS) due to incorporation of EAS and RAS in IA. The integration of these algorithms and statistical tools, result in six variants of ant algorithms like Performance Linked Influential Elitist Ant Sytem - Mid-Range(PLIEASMR), Performance Linked Influential Elitist Ant Sytem - Mean(PLIEASM), Performance Linked Influential Elitist Ant Sytem - Median (PLIEASMed), Performance Linked Influential Rank Ant Sytems - Mid-Range(PLIRASMR), Performance Linked Influential Rank Ant Sytems - Mean(PLIRASM) and Performance Linked Influential Rank Ant Sytems - Median(PLIRASMed).

IA Specification for PLIEAS

The algorithmic specification for PLIEAS suggests the reinforcement of IA paths and is given by the expression

$$e^{*}.Q^{*}/L \tag{2.6}$$

where e^* is the number of IA selected by a classifier in a given iteration, Q^* is algorithmic constant and L is the best tour length in the performing ant's tour list.

IA Specification for PLIRAS

The algorithmic specification for PLIRAS suggests to rank the performance of IA and accordingly reinforce their traveled paths. It is given by the expression

$$\Delta\tau_{ij}^{k} = \begin{cases} Q^{*}\cdot(e^{*}-k)/L_k & \text{if (i, j)} \in k^{th} \text{ performing ant's tour list.} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where L_k is the tour length of k^{th} performing ant and k is the ranking index.

It should be noted that elitist selection mechanism introduces the dynamicity in search space by varying the number of elite ants across the iterations, unlike traditional EAS and RAS, where in, the number of elite ants were fixed as a part of parameter settings.

2.2.6 Punishment mechanism in PLIEAS and PLIRAS

The PLIEAS and PLIRAS algorithms can be extended by incorporating the punishment features into it. In the process, non performing ants will be punished by removing certain amount of pheromone on the traveled path. The quantity of pheromone to be removed will be specified by punished IA specification. These new algorithms here onwards will be named as Punished PLIEAS (PPLIEAS) and Punished PLIRAS (PPLIRAS). The generic punishment feature for IA is given by the equation:

$$\tau_{ij} = \tau_{ij} - \Delta\tau_{ij}^{*} \tag{2.8}$$

where $\Delta\tau_{ij}^{*}$ is

$$\Delta\tau_{ij}^{*} = \sum_{k=1}^{l^{*}} \Delta\tau_{ij}^{k}$$

where l^{*} is the number of punished ants and is given by the expression $l^{*} = m - e^{*}$.

The generic pseudo-code for Punished Performance Linked Elitist ACO is given by Algorithm 7:

The Punished Performance Linked Elitist ACO has an additional procedure *UpdateNonEliteAntPheromone* to reflect the punishment mechanism. The procedure implements a code to remove certain amount of pheromone on non elite paths traveled by non performing ants.

Punished IA Specification for PPLIEAS

The punished IA specification for PPLIEAS specifies to decrease the quantity of pheromone trial proportional to the quality of solution found on a non elite paths.

Algorithm 7 Punished Performance Linked Elitist Ant Colony Optimization

Initialize the pheromone values.
while termination conditions not met **do**
  START ScheduleActivities
  ConstructAntsSolutions
  SelectEliteAnts
  UpdateAntPheromone
  UpdateEliteAntPheromone
  UpdateNonEliteAntPheromone
  DeamonActions
  END ScheduleActivities
end while

$\Delta\tau_{ij}^k$ for PPLIEAS is given by the equation:

$$\Delta\tau_{ij}^k = \begin{cases} l^{*} \cdot Q^{*}/L_k & \text{if (i, j)} \in k^{th} \text{ non performing ant's tour list} \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

Punished IA Specification for PPLIRAS

The punished IA specification for PPLIRAS specifies that non elite paths are ranked according to their performances and then proportionately decreased. $\Delta\tau_{ij}^k$ for PPLIRAS is given by the equation:

$$\Delta\tau_{ij}^k = \begin{cases} Q^{*} \cdot (l^{*} - k)/L_k & \text{if (i, j)} \in k^{th} \text{ non performing ant's tour list} \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

The equations (2.9) and (2.10) suggest that pheromone trail removal will be proportional to the quality of solution found.

2.2.7 Experimental Results and Performance Analysis

Parameter Settings

Inorder to assess the performance of the above algorithms, parameters α, β were varied from 1 to 5 and ρ was varied from 0.7 to 1.0. The algorithms were executed 10 times independently by considering some of the datasets available in the TSPLIB (see Table 2.1). The maximum number of iterations was set to $1,00,000$.

Computational Results and Comparative Analysis for IA

Table 2.4 provides the comparative results for PLIEAS and PLIRAS algorithms incorporated with statistical tool (Mean, Median and Mid-Range tour selection). These variants were compared with MMAS+IB+PTS for best solution, average solution and the percentage of deviation from the optimal solution. The average solution was computed using the best solutions of last 50 iterations. Table 2.4 shows that PLIEASM provides best solution for att48 dataset with deviation of 0.04% and PLIEASMed provides best solution for st70 dataset with deviation of 0.05%. Similarly, PLIRASM provides best solution for Kroa100, lin318 with observed deviation of 0.01% and PLIRASMed provides best results for Kroa100 and Kroa200 with no deviation. The solutions provided by MRTS deviates more from the optimal solution compared to other tour selection mechanism for both PLIEAS and PLIRAS algorithms and it can be attributed to updation of not so promising paths. The algorithmic simulation suggests that median function takes lesser number of iterations to find optimal solution than the mean function. Another interesting observation is that, average solutions of proposed algorithms for most of the datasets have larger deviation from the optimal solution, indicating lack of focus to concentrate on promising region of search space.

Table 2.4: Performance comparasion of Influential Ants on various datasets.

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	MMAS+IB+PTS	2022.1 (0.1%)	2025.3 (0.26%)
	PLIEASMR	2039.4 (0.96%)	2055.6 (1.76%)
	PLIEASM	2034.7 (0.72%)	2046.9 (1.33%)
bays29	PLIEASMed	**2021.7 (0.08%)**	2047.3 (1.35%)
	PLIRASMR	2044.4 (1.20%)	2059.5 (1.95%)
	PLIRASM	2040.8 (1.02%)	2055.9 (1.77%)
	PLIRASMed	2042.2 (1.09%)	2057.8 (1.87%)

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
att48	MMAS+IB+PTS	10634.4 (0.06%)	10640.8 (0.12%)
	PLIEASMR	10821.8 (1.82%)	10892.4 (2.48%)
	PLIEASM	**10632.8 (0.04%)**	10695.1 (0.63%)
	PLIEASMed	10638.2 (0.09%)	10702.7 (0.7%)
	PLIRASMR	10768.9 (1.32%)	10844.7 (2.03%)
	PLIRASM	10656.8 (0.27%)	10690.5 (0.58%)
	PLIRASMed	10643.8 (0.14%)	10697.4 (0.65%)
eil51	MMAS+IB+PTS	426.2 (0.04%)	427.8 (0.43%)
	PLIEASMR	438.4 (2.42%)	448.3 (4.74%)
	PLIEASM	426.5 (0.11%)	442.6 (3.41%)
	PLIEASMed	426.8 (0.18%)	433.2 (1.21%)
	PLIRASMR	436.2 (1.91%)	444.9 (3.94%)
	PLIRASM	434.6 (1.54%)	442.7 (3.43%)
	PLIRASMed	431.6 (0.84%)	439.7 (2.73%)
st70	MMAS+IB+PTS	675.5 (0.07%)	680.3 (0.78%)
	PLIEASMR	712.7 (5.58%)	730.4 (8.20%)
	PLIEASM	678.5 (0.51%)	710.1 (5.2%)
	PLIEASMed	**675.4 (0.05%)**	698.3 (3.45%)
	PLIRASMR	685.3 (1.52%)	710.7 (5.28%)
	PLIRASM	678.5 (0.51%)	689.4 (2.13%)
	PLIRASMed	677.2 (0.32%)	686.4 (1.68%)
eil76	MMAS+IB+PTS	538.5 (0.09%)	539.9 (0.35%)
	PLIEASMR	552.6 (2.71%)	582.4 (8.25%)
	PLIEASM	545.7 (1.43%)	551.4 (2.49%)
	PLIEASMed	547.6 (1.78%)	561.2 (4.31%)
	PLIRASMR	554.3 (3.02%)	575.7 (7%)
	PLIRASM	542.4 (0.81%)	549.8 (2.19%)
	PLIRASMed	540.3 (0.42%)	548.7 (1.98%)

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	MMAS+IB+PTS	21285.4 (0.01%)	21336.9 (0.26%)
	PLIEASMR	21890.8 (3.28%)	22140.7 (4.03%)
	PLIEASM	21540.6 (1.21%)	21598.6 (1.48%)
Kroa100	PLIEASMed	21385.7 (0.48%)	21456.9 (0.82%)
	PLIRASMR	21780.4 (2.34%)	21930.7 (3.04%)
	PLIRASM	**21284.6 (0.01%)**	21436.3 (0.72%)
	PLIRASMed	**21283.8 (0%)**	21385.6 (0.48%)
	MMAS+IB+PTS	29372.2 (0.01%)	29385.8 (0.06%)
	PLIEASMR	31468.9 (7.15%)	31790.5 (8.24%)
	PLIEASM	29640.6 (0.92%)	29689.9 (1.09%)
kroa200	PLIEASMed	29536.7 (0.57%)	29590.3 (0.75%)
	PLIRASMR	29840.6 (1.60%)	29994.8 (2.13%)
	PLIRASM	29390.9 (0.07%)	29569.4 (0.68%)
	PLIRASMed	**29370.3 (0%)**	29480.8 (0.38%)
	MMAS+IB+PTS	42035.7 (0.01%)	42055.8 (0.06%)
	PLIEASMR	44896.5 (6.82%)	45218.7 (7.58%)
	PLIEASM	43220.6 (2.83%)	43312.7 (3.05%)
lin318	PLIEASMed	42870.4 (2%)	42910.3 (2.09%)
	PLIRASMR	43723.6 (4.03%)	43890.4 (4.42%)
	PLIRASM	**42034.5 (0.01%)**	42392.3 (0.86%)
	PLIRASMed	42137.2 (0.25%)	42297.8 (0.63%)

Parameter Sensitivity Analysis for IA

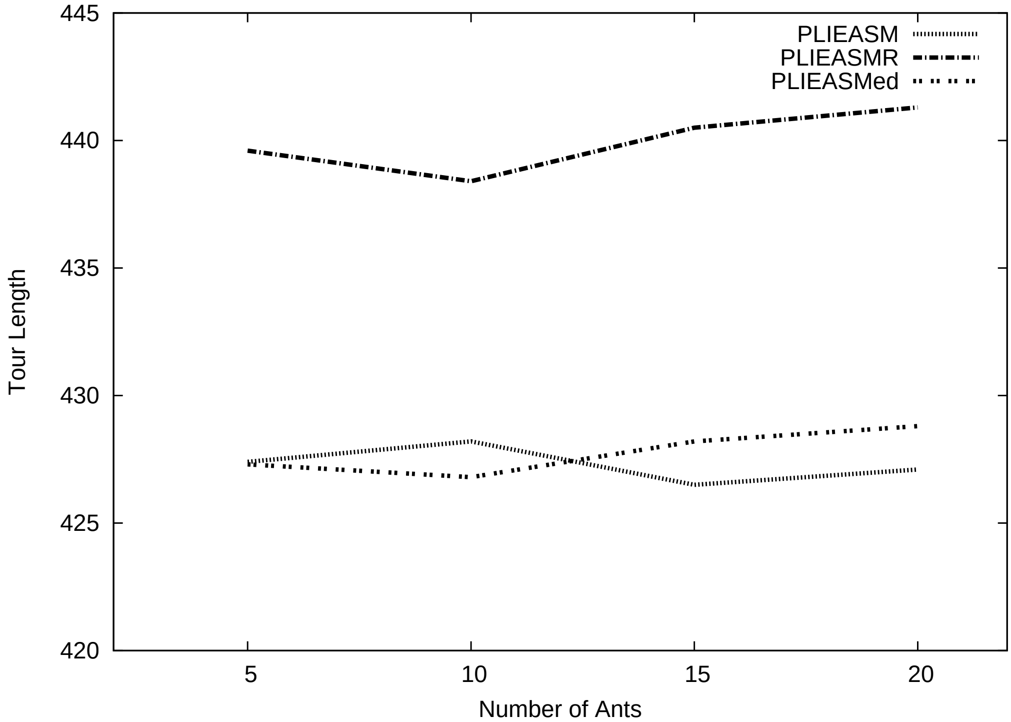
Figure 2.4 shows the performance of PLIEAS variants for eil51 dataset. In general, performance of MRTS variants is inferior to MTS and MeTS for varying ACO parameters. It can be observed from Figure 2.4(a) that PLIEASMed variant performance is better for smaller ants population and PLIEASM variant performs well for larger ants population. Figure 2.4(b) shows the sensitiveness of PLIEAS variant for varying trial strength. The PLIEASMed variant exhibits lesser variation compared to other variants and provides best result for ρ=0.9. The graph also reveals that PLIEASMed variant performs better for higher trial strength and PLIEASM for trial strength with ρ=0.8.

Figure 2.5 shows the performance of PLIRAS variants for st70 dataset. Figure 2.5(a) reveals that performance of PLIEASM is better for smaller ants population in the range $5 \leq n \leq 10$ and for PLIEASMed in the range of $10 \leq n \leq 15$. The PLIEASM and PLIEASMed variants provide best result for n=10. Figure 2.5(b) reveals some interesting facts. Firstly, higher pheromone evaporation rate has a bad effect on the performance of PLIEASMed and PLIEASMR variants. Secondly, PLIEASMed variant shows sharp improvement in the observed tour lengths for $\rho \geq 0.85$ with the observed best result for ρ=0.95.

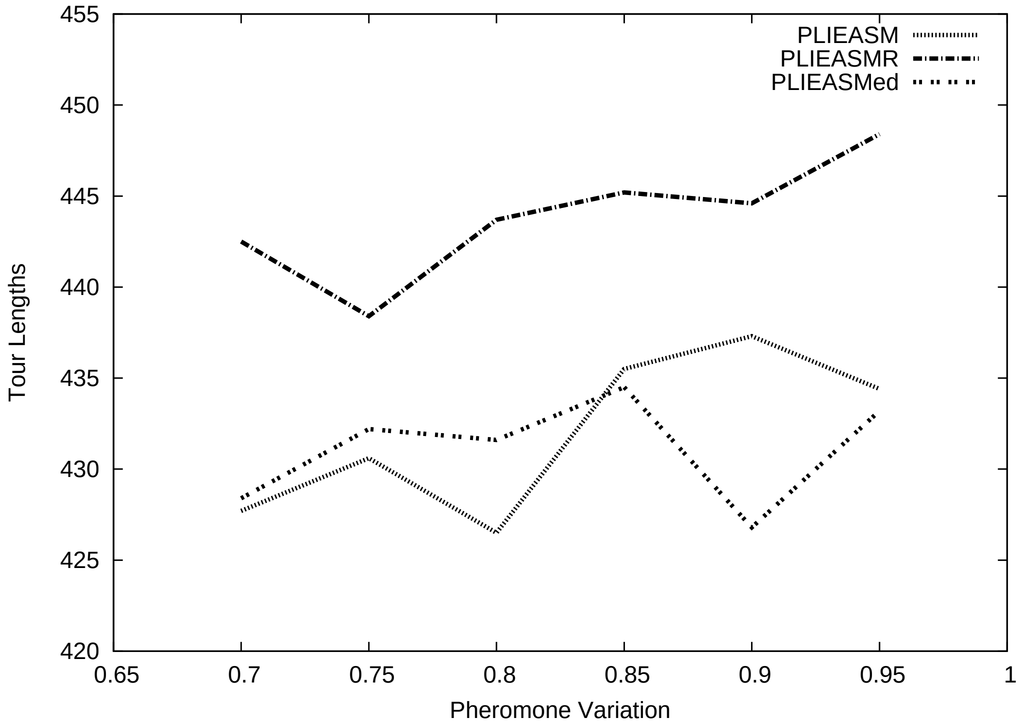
Table 2.5 reports the observed parameter values, when the optimal solutions were obtained for PPLIEAS and PPLIRAS algorithms. The parameter values are specified in range bounds for all the algorithm variants to indicate the most common observations for the datasets under consideration. It can be observed that parameter α exhibits larger variation in values from 1 to 4 and β usually varies from 1 to 3. Similar observation can be made for ρ. For the MRTS variants, better solution quality was obtained for lower trial values and it varies from $0.7 \leq \rho \leq 0.82$. For MTS variants, better result was obtained for higher trial values that varies from $0.86 \leq \rho \leq 0.95$ and MeTS variants exhibit a larger variation in the range of $0.76 \leq \rho \leq 0.99$.

Table 2.5: Parameter details for PLIEAS and PLIRAS

PLIEASMR	$2 \leq \alpha \leq 4$	$1 \leq \beta \leq 3$	0.7 - 0.79
PLIEASM	$1 \leq \alpha \leq 3$	$2 \leq \beta \leq 3$	0.86 - 0.95
PLIEASMed	$3 \leq \alpha \leq 4$	$1 \leq \beta \leq 2$	0.92 - 0.99
PLIRASMR	$2 \leq \alpha \leq 3$	$2 \leq \beta \leq 4$	0.76 - 0.82
PLIRASM	$1 \leq \alpha \leq 3$	$2 \leq \beta \leq 3$	0.89 - 0.95
PLIRASMed	$2 \leq \alpha \leq 4$	$1 \leq \beta \leq 3$	0.76 - 0.93

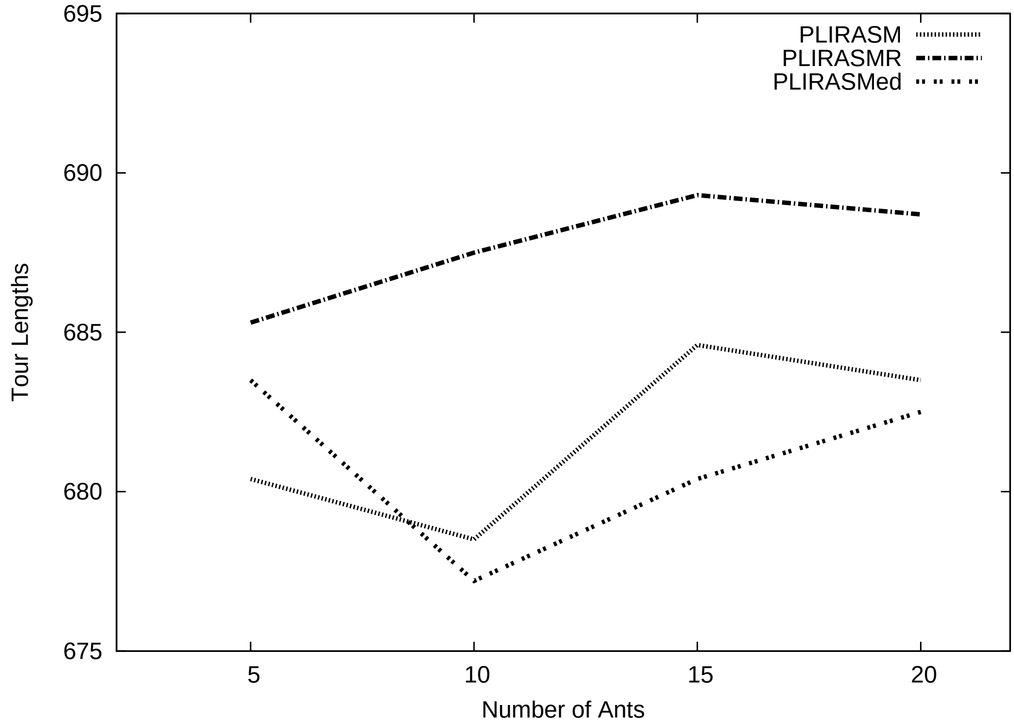


(a) Variation in number of ants

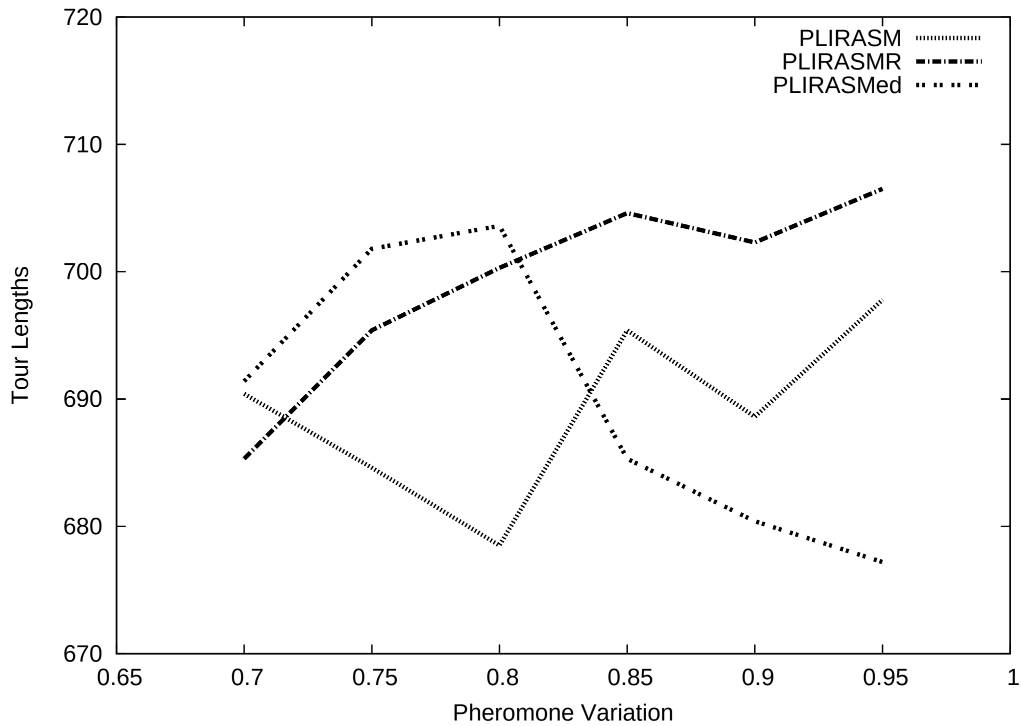


(b) Variation in pheromone trial

Figure 2.4: Performance comparison of PLIEAS variants for eil51 dataset.



(a) Variation in number of ants



(b) Variation in pheromone trial

Figure 2.5: Performance comparision of PLIRAS variants for st70 dataset.

Computational Results and Comparative Analysis for Punished IA

Table 2.6 shows the incorporation of punishment mechanism to IA. On comparison with Table 2.4, it can be observed that punishment mechanism improvises the solution and also provides the best optimal solution for some of the datasets. Table 2.6 reveals that PPLIEASM provides best solution for att48 dataset with the deviation of 0.03% and for st70 dataset with the deviation of 0.04%. The PPLIEASMed provides better solution for bays29, eil51 dataset with deviation in observed solution 0.07%, 0.02% respectively and PPLIRASMed provides best solution for Kroa100, Kroa200 with no deviation. Similar observation can be made for PPLIRASM that exhibits no deviation for lin318 dataset. The punishment mechanism improvises the average solution for most of the datasets under consideration, demonstrating the ability of algorithms in restricting the search in promising area of search space.

Table 2.6: Performance comparision of Punished Influential Ants on various datasets.

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	MMAS+IB+PTS	2022.1 (0.1%)	2025.3 (0.26%)
	PPLIEASMR	2053.5 (1.65%)	2082.1 (3.07%)
	PPPLIEASM	2022.4 (0.11%)	2027.8 (0.38%)
bays29	PPLIEASMed	**2021.5 (0.07%)**	**2023.6 (0.17%)**
	PPLIRASMR	2052.8 (1.62%)	2074.4 (2.69%)
	PPLIRASM	2038.9 (0.93%)	2044.9 (1.23%)
	PPLIRASMed	2034.7 (0.72%)	2044.6 (1.21%)
	MMAS+IB+PTS	10634.4 (0.06%)	10640.8 (0.12%)
	PPLIEASMR	10730.5 (0.96%)	10834.4 (1.94%)
	PPLIEASM	**10632.2 (0.03%)**	10645.2 (0.16%)
att48	PPLIEASMed	**10630.4 (0.02%)**	**10638.8 (0.1%)**
	PPLIRASMR	10656.7 (0.27%)	10712.4 (0.79%)

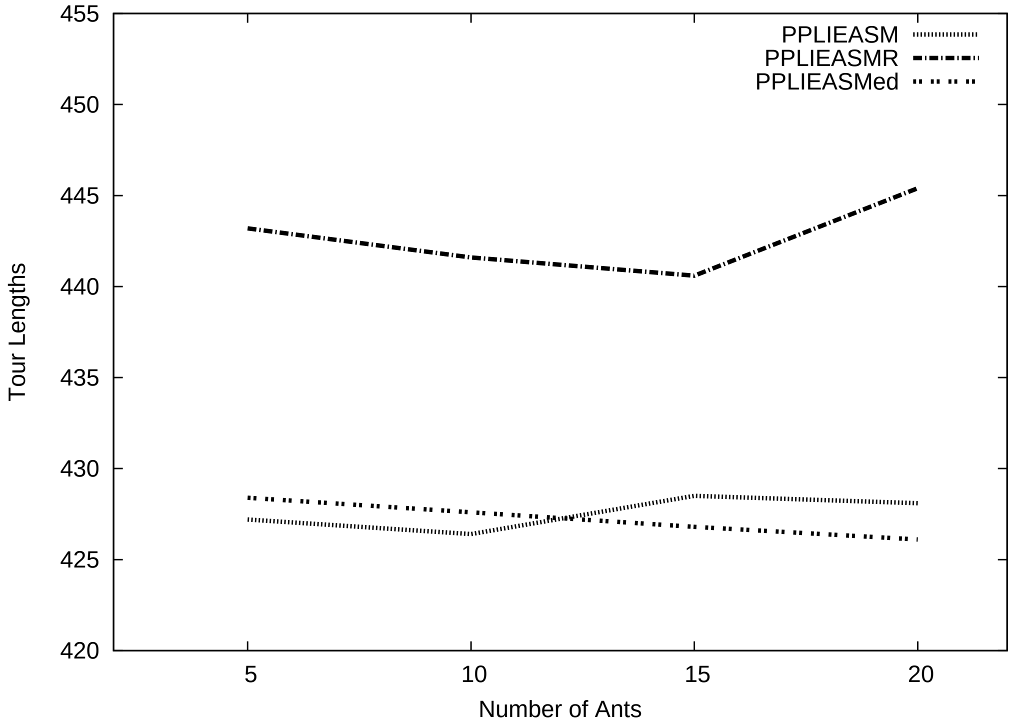
Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	PPLIRASM	10643.2 (0.14%)	10668.8 (0.38%)
	PPLIRASMed	10645.9 (0.16%)	10674.6 (0.43%)
eil51	MMAS+IB+PTS	426.2 (0.04%)	427.8 (0.43%)
	PPLIEASMR	440.6 (2.94%)	455.3 (6.37%)
	PPLIEASM	426.4 (0.09%)	434.3 (1.47%)
	PPLIEASMed	**426.1 (0.02%)**	**427.4 (0.32%)**
	PPLIRASMR	440.5 (2.92%)	452.3 (5.67%)
	PPLIRASM	432.7 (1.09%)	438.9 (3.02%)
	PPLIRASMed	430.8 (0.65%)	437.3 (2.65%)
st70	MMAS+IB+PTS	675.5(0.07%)	680.3(0.78%)
	PPLIEASMR	698.4 (3.46%)	720.3 (6.71%)
	PPLIEASM	**675.3 (0.04%)**	**678.6 (0.53%)**
	PPLIEASMed	680.8 (0.85%)	685.3 (1.52%)
	PPLIRASMR	691.5 (2.44%)	715.6 (6.01%)
	PPLIRASM	676.5 (0.22%)	684.5 (1.40%)
	PPLIRASMed	**675.4 (0.05%)**	682.7 (1.14%)
eil76	MMAS+IB+PTS	538.5 (0.09%)	539.9 (0.35%)
	PPLIEASMR	561.4 (4.34%)	584.3 (8.60%)
	PPLIEASM	541.4 (0.63%)	548.4 (1.93%)
	PPLIEASMed	545.6 (1.41%)	555.4 (3.23%)
	PPLIRASMR	548.5 (2%)	564.3 (4.88%)
	PPLIRASM	541.4 (0.63%)	545.6 (1.41%)
	PPLIRASMed	538.8 (0.14%)	543.7 (1.05%)
Kroa100	MMAS+IB+PTS	21285.4 (0.01%)	21336.9 (0.26%)
	PPLIEASMR	21780.4 (2.34%)	21867.3 (2.75%)
	PPLIEASM	21321.6 (0.18%)	21375.1 (0.43%)
	PPLIEASMed	21330.7 (0.22%)	21367.8 (0.4%)
	PPLIRASMR	21610.6 (1.54%)	21688.7 (1.91%)

Datasets	Algorithms	Best (Std Dev)	Average (Std Dev)
	PPLIRASM	21286.3 (0.02%)	**21295.7 (0.06%)**
	PPLIRASMed	**21282.7 (0%)**	**21288.4 (0.03%)**
kroa200	MMAS+IB+PTS	29372.2 (0.01%)	29385.8 (0.06%)
	PPLIEASMR	30850.7 (5.04%)	31224.7 (6.32%)
	PPLIEASM	29540.8 (0.58%)	29588.4 (0.75%)
	PPLIEASMed	29444.8 (0.26%)	29489.2 (0.41%)
	PPLIRASMR	29664.8 (1.01%)	29710.7 (1.16%)
	PPLIRASM	29446.5 (0.26%)	29486.5 (0.40%)
	PPLIRASMed	**29368.5 (0.0%)**	**29380.8 (0.04%)**
lin318	MMAS+IB+PTS	42035.7 (0.01%)	42055.8 (0.06%)
	PPLIEASMR	44219.6 (5.12%)	44870.4 (6.76%)
	PPLIEASM	42780.5 (1.78%)	42932.7 (2.15%)
	PPLIEASMed	42540.8 (1.21%)	42624.3 (1.41%)
	PPLIRASMR	43879.5 (4.40%)	43964.3 (4.60%)
	PPLIRASM	**42033.2 (0%)**	**42042.9 (0.03%)**
	PPLIRASMed	42045.7 (0.03%)	42094.3 (0.15%)

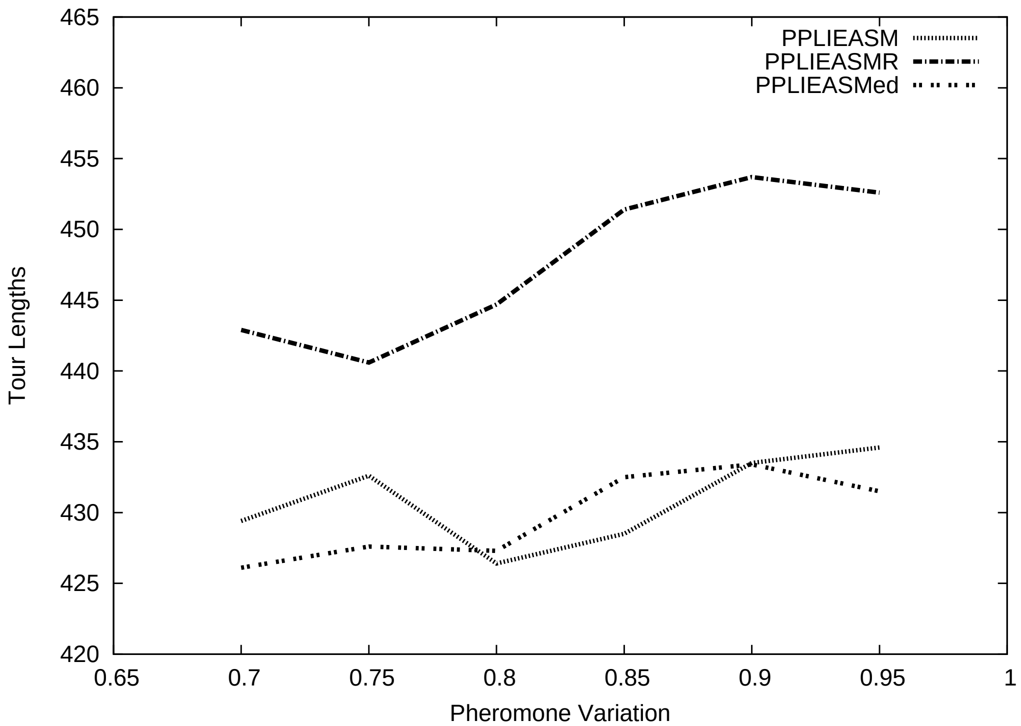
Parameter Sensitivity Analysis for Punished IA

Figure 2.6 shows the performance of PPLIEAS variants for eil51 dataset. Figure 2.6(a) reveals the behavior of ants for varying population size. It can be observed that PPLIEASMed variant shows an improvement in result with the increase in ants population. Similarly, performance of PPLIEASM variant is slightly better for smaller ants population compared to larger ants population. The PPLIEASM and PPLIEASMed provide best result for ρ equal to 10 and 20 respectively. Figure 2.6(b) shows that PPLIEASMed exhibit lesser variation compared to other variants and provides best result for lower trial strength of ρ=0.7. Similarly, PPLIEASM provides the best result for ρ=0.8.

Figure 2.7 shows the performance of PPLIRAS variants for st70 dataset. Figure 2.7(a) reveals that both the variants i.e., PPLIRASM and PPLIRASMed perform



(a) Variation in number of ants



(b) Variation in pheromone trial

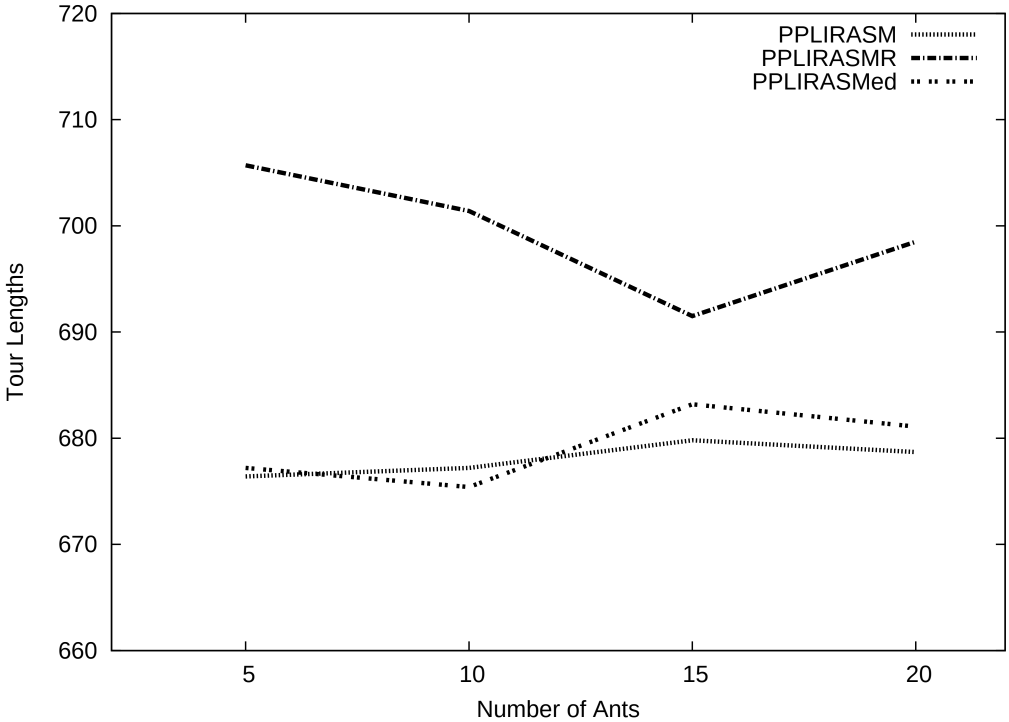
Figure 2.6: Performance comparison of PPLIEAS variants for eil51 dataset.

better for smaller ants population and quite a contrasting observation for PPLIRASMR, which performs well for larger ants population. Similarly, Figure 2.7(b) reveals that PPLIRASM and PPLIRASMed variants do not show any definite trend for the varying trail strength and it is hard to make any meaningful conclusion about the effect of varying trial strength. The PPLIRASM and PPLIRASMed variants provide best result for ρ=0.8.

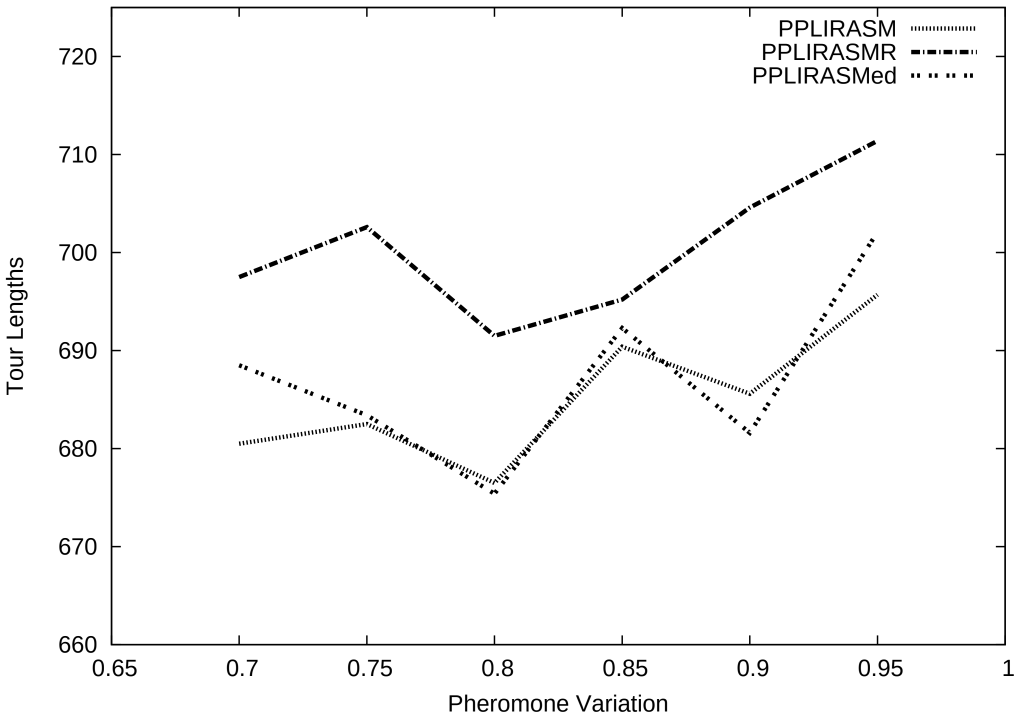
Table 2.7 reports the observed parameter values, when the optimal solutions were obtained for PPLIEAS and PPLIRAS algorithms. The parameter values are specified in range bounds for all the algorithm variants to indicate the most common observations for the datasets under consideration.

Table 2.7: Parameter details for PPLIEAS and PPLIRAS

PPLIEASMR	$1 \leq \alpha \leq 3$	$2 \leq \beta \leq 3$	0.89 - 0.96
PPLIEASM	$2 \leq \alpha \leq 3$	$2 \leq \beta \leq 4$	0.73 - 0.82
PPLIEASMed	$3 \leq \alpha \leq 5$	$1 \leq \beta \leq 3$	0.85 - 0.95
PPLIRASMR	$1 \leq \alpha \leq 2$	$1 \leq \beta \leq 3$	0.76 - 0.82
PPLIRASM	$2 \leq \alpha \leq 4$	$2 \leq \beta \leq 4$	0.92 - 0.99
PPLIRASMed	$2 \leq \alpha \leq 4$	$2 \leq \beta \leq 3$	0.85 - 0.96



(a) Variation in number of ants



(b) Variation in pheromone trial

Figure 2.7: Performance comparison of PPLIRAS variants for st70 dataset.

Table 2.8: Average Ants Selection for Influential Ants

	PLIEASMR	PLIEASM	PLIEASMed	PLIRASMR	PLIRASM	PLIRASMed
baysg29	7.14(10)	6.88(10)	6.37(10)	7.36(10)	7.20(10)	7.26(10)
att48	7.52(10)	6.18(10)	6.47(10)	7.36(10)	6.52(10)	6.23(10)
eil51	15.42(20)	12.36(20)	12.54(20)	15.18(20)	14.83(20)	14.57(20)
st70	8.16(10)	6.44(10)	6.12(10)	7.18(10)	6.38(10)	6.24(10)
eil76	7.66(10)	6.85(10)	7.14(10)	7.84(10)	6.83(10)	6.57(10)
Kroa100	15.16(20)	13.36(20)	12.58(20)	14.94(20)	12.44(20)	12.21(20)
Kroa200	26.72(30)	20.82(30)	18.48(30)	21.76(30)	18.78(30)	18.15(30)
lin318	27.44(30)	23.13(30)	21.64(30)	23.95(30)	18.33(30)	18.68(30)

Table 2.9: Average Ants Selection for Punished Influential Ants

	PPLIEASMR	PPLIEASM	PPLIEASMed	PPLIRASMR	PPLIRASM	PPLIRASMed
baysg29	6.58(10)	5.14(10)	5.07(10)	6.53(10)	5.72(10)	5.53(10)
att48	6.15(10)	5.03(10)	5.01(10)	5.26(10)	5.18(10)	5.23(10)
eil51	13.84(20)	10.48(20)	10.36(20)	14.26(20)	12.20(20)	11.78(20)
st70	6.87(10)	5.18(10)	5.55(10)	6.56(10)	5.44(10)	5.21(10)
eil76	7.24(10)	5.45(10)	5.52(10)	5.70(10)	5.37(10)	5.11(10)
Kroa100	13.76(20)	10.46(20)	10.62(20)	13.56(20)	10.18(20)	10.05(20)
Kroa200	22.68(30)	16.16(30)	15.68(30)	17.44(30)	15.84(30)	15.28(30)
lin318	22.94(30)	17.74(30)	17.32(30)	21.44(30)	15.38(30)	15.47(30)

Selection Analysis of IA and Punished IA

Table 2.8 provides the detail of average number of ants selected for second time updation and the total number of ants considered for the experimentation. In MRTS mechanism, 70-80% of the ants in the system were selected for additional reinforcement and in case of MTS and MeTS it was 60-70%. The selection of larger number of ants in MRTS indicates that the algorithm spends more time in exploring rather than exploiting the search space to look for optimal solution. Table 2.9 shows the details of average ants usage for punished influential ants. The MRTS variant selects 60-70% and MTS, MeTS select in the range of 50-60% of ants in the system for additional reinforcement. On comparison of Table 2.8 with Table 2.9, it can be observed that punishment mechanism selects fewer number of ants compared to non-punished mechanism. This demonstrates that punishment mechanism has succeeded in restricting the search process in promising region of search space.

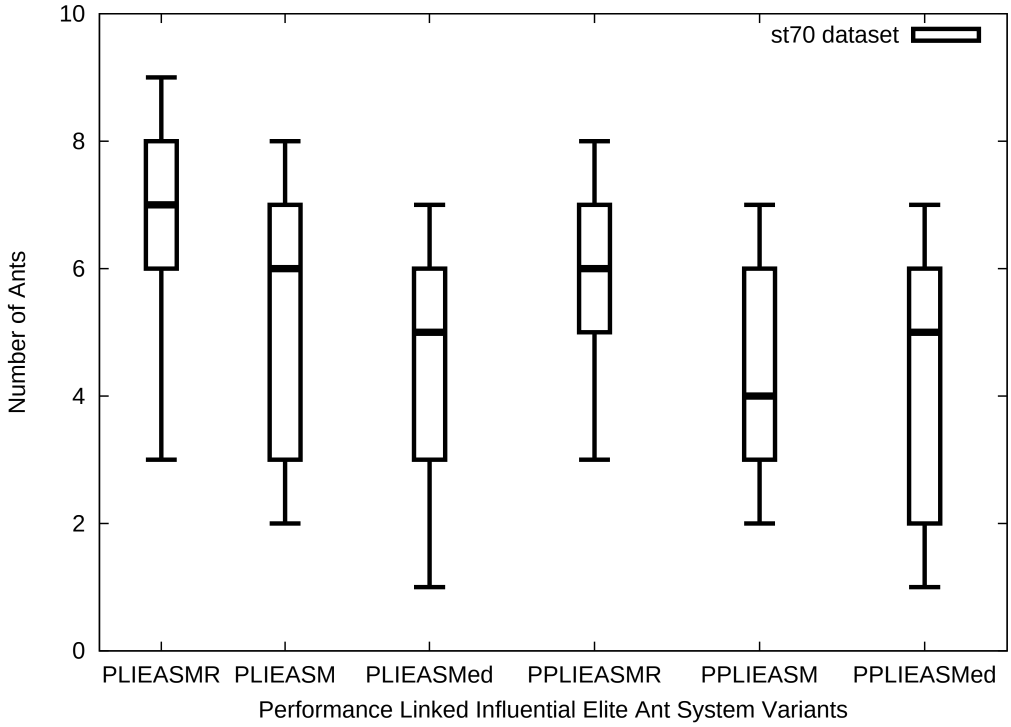
Distribution Analysis of IA and Punished IA

Further analysis on ant's selection mechanism was done by plotting the Box and Whisker graphs, which provide the details of ants selection distribution in the form of five number summary. Figure 2.8 and Figure 2.9 show spread in the distribution of selected ants during algorithm execution for st70 and kroa100 datasets. Some interesting observations have been made and a comparative analysis of IA variants from statistical tool perspective were done. Figure 2.8(a) and Figure 2.8(b) show that Inter Quartile Range (IQR) was comparatively smaller for MRTS and skewed towards upper whisker. The MRTS incorporated IA non punished variants have 6 to 7 ants respectively at 50% (median) observation indicating that, most of the time, algorithm selects larger number of ants. However, a better spread is observed in MTS and MeTS variants and the observed median value is in the range of 5 to 6 ants. The punishment mechanism provides better results compared to non punishment mechanism as shown in Figure 2.9. The punishment mechanism is characterized by selection of fewer number of ants. In case of MRTS the observed median value is 6 and for MeTS, MTS it ranges from 4 to 5. In general, it can be concluded that a better spread in selection indicates better exploration and exploitation of search space, that leads to better solution.

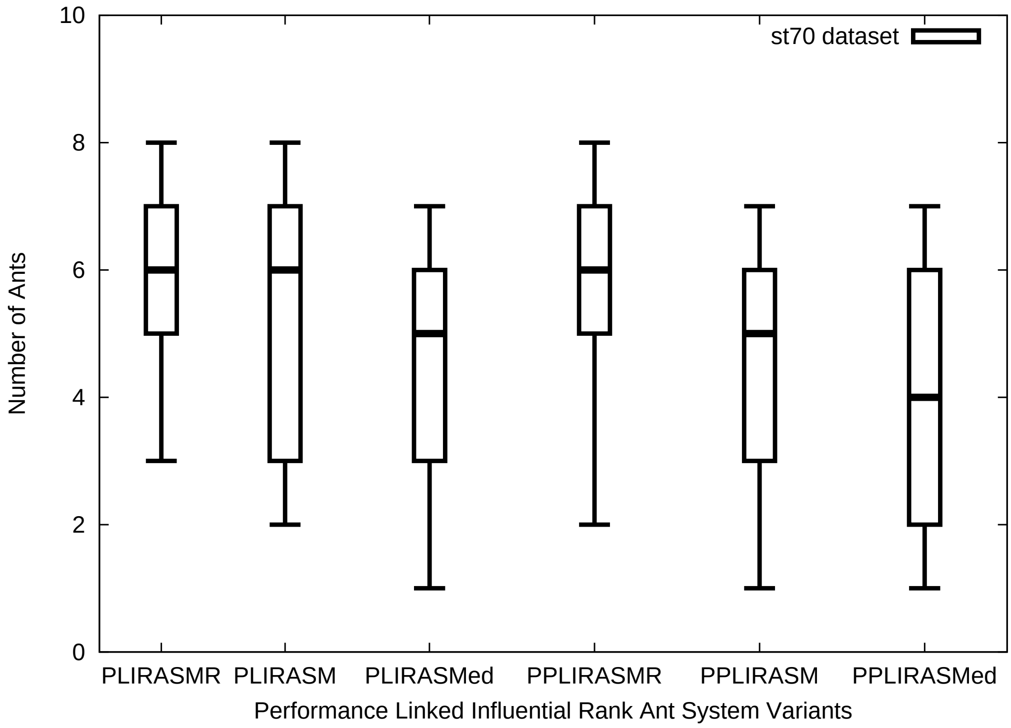
2.3 Concluding Remarks

This chapter outlined the significance of influential ants and punishment mechanism for ACO algorithms. The methods proposed here showed superior performance to that of MMAS for most of the datasets. Thus, it is clear that such an unsupervised elitism will lead to better quality of solution without unduly compromising on convergence. As a result, an adequate non-static elitism imparts ants diversity, thereby improving the performance subject to the problem specification.

Furthermore, a feasibility of these algorithms for real world applications has been demonstrated by investigating the performance on train scheduling problem in the Chapter 5.

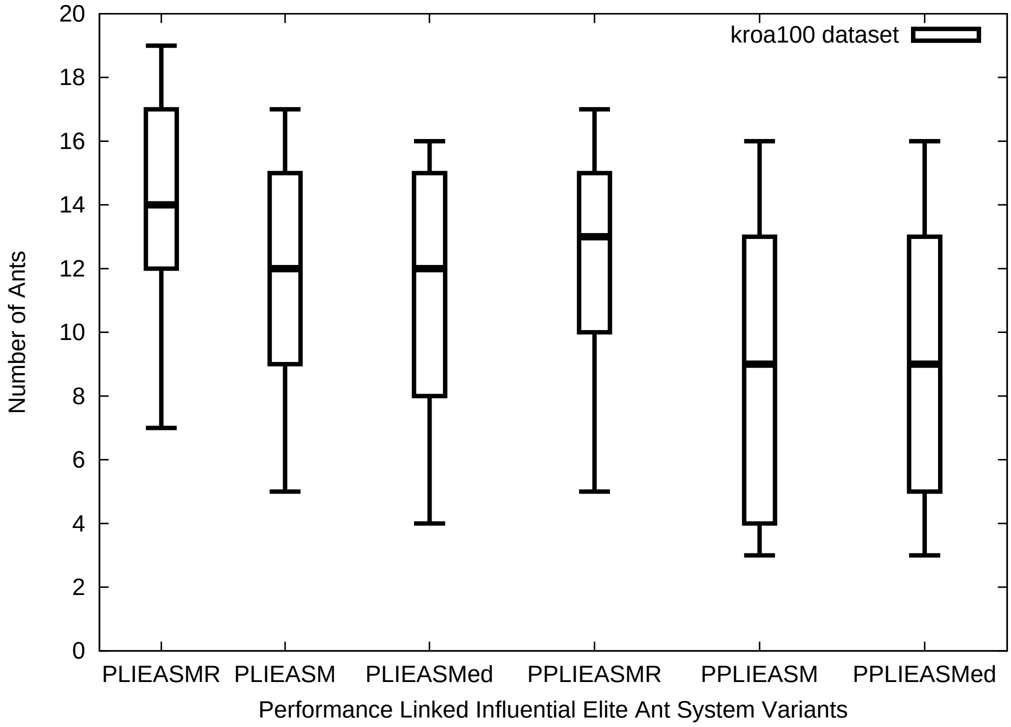


(a) Comparative Analysis of Median Variance for Punished and Non-Punished PLIEAS

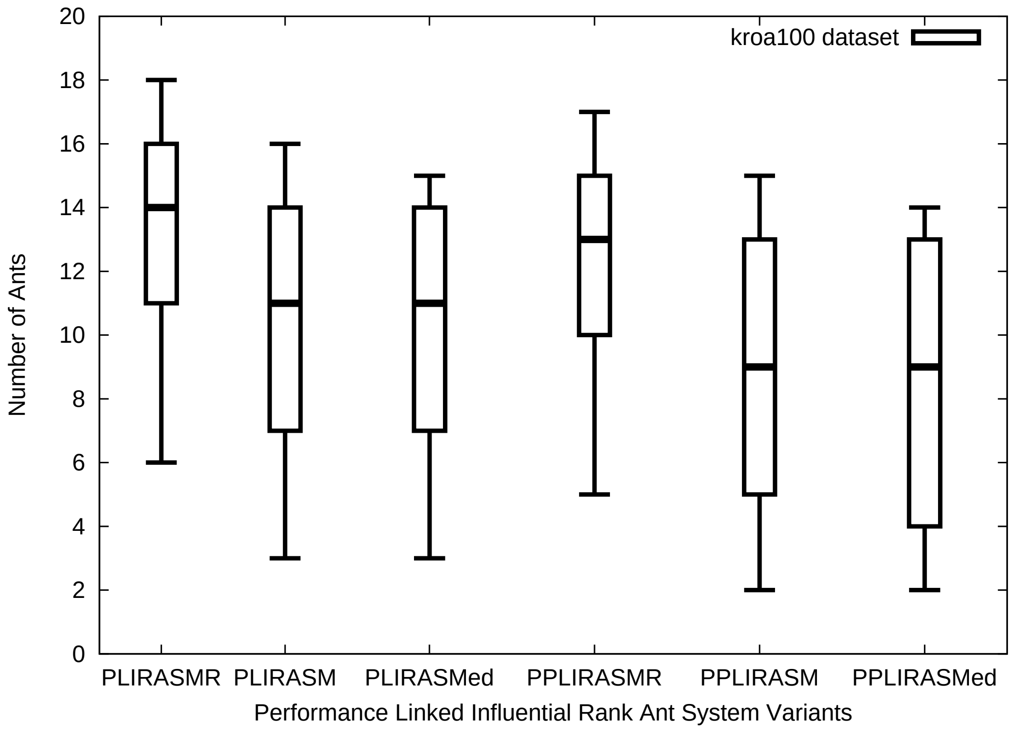


(b) Comparative Analysis of Median Variance for Punished and Non-Punished PLIRAS

Figure 2.8: Box and Whisker plots showing the distribution of ants selection for additional reinforcement.



(a) Comparative Analysis of Median Variance for Punished and Non-Punished PLIEAS



(b) Comparative Analysis of Median Variance for Punished and Non-Punished PLIRAS

Figure 2.9: Box and Whisker plots showing the distribution of ants selection for additional reinforcement.