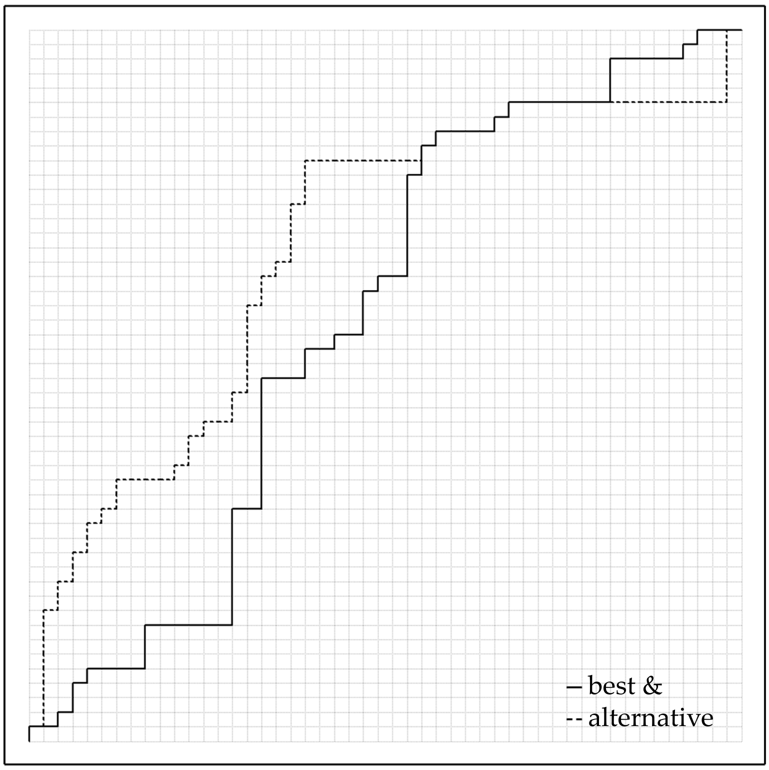
Jörg Sameith

Penalty Methods

Generating Alternative Solutions for Discrete Optimization Problems with Uncertain Data



3-Hirn-Verlag

Jörg Sameith

Penalty Methods
—
Generating Alternative Solutions for Discrete Optimization Problems with Uncertain Data

3-Hirn-Verlag

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Sameith, Jörg:
Penalty Methods — Generating Alternative Solutions for Discrete Optimization Problems with Uncertain Data. 3-Hirn-Verlag, Lage, 2005.
Zugl.: Jena, Univ., Diss., 2005
ISBN 3-937583-06-8

Bestellung beim Verlag

3-Hirn-Verlag
Schubertstr. 16
32791 Lage
Tel./Fax.: (05232) 929073
Webseite: www.3-hirn-verlag.de

3-Hirn-Verlag

ISBN 3-937583-06-8

Penalty Methods

—

Generating Alternative Solutions for Discrete Optimization Problems with Uncertain Data

Dissertation zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena
von Diplom-Wirtschaftsmathematiker Jörg Sameith
geboren am 7.1.1978 in Bad Salzungen

Erster Gutachter	:	Prof. Dr. Ingo Althöfer
Zweiter Gutachter	:	Prof. Dr. Richard Lorentz
Tag der		
letzten Prüfung des Rigorosums	:	08.11.2005
öffentlichen Verteidigung	:	24.11.2005

Contents

Zusammenfassung – Abstract in German

Die vorliegende Dissertation untersucht **“Das Generieren von Alternativ-Lösungen für diskrete Optimierungs-Probleme mit unsicheren Daten”**.

Das Hauptaugenmerk gilt dabei Optimierungs-Problemen vom Summen-Typ, z.B. Kürzeste-Wege-Problem, Rundreiseproblem, Zuordnungsproblem, Rucksackproblem. Bei diesen Problemen besteht eine zulässige Lösung aus einer Menge von verschiedenen Teilkomponenten (z.B. Kanten eines Graphen), denen bestimmte Kosten zugeordnet sind. Der Zielfunktionswert einer zulässigen Lösung ist dann die Summe der Kosten aller benutzten Teilkomponenten. Gesucht werden Lösungen mit kleiner Kostensumme.

Bevor ein Optimierungs-Problem aus der Praxis mit Methoden der mathematischen Optimierung bearbeitet werden kann, muss es auf die Ebene eines mathematischen Modells abstrahiert werden. Dabei müssen häufig vereinfachende Annahmen (z.B.: zeitlich variierende Problem-Daten werden geschätzt und fixiert, komplexe Einfluss-Faktoren werden vernachlässigt, ...) getroffen werden, so dass das mathematische Modell das ursprüngliche Optimierungs-Problem nicht perfekt widerspiegelt. Dann kann zwar häufig eine sehr gute oder optimale Lösung für das Modell berechnet werden, die Güte ist aber in der Realisierung in der Praxis nicht zwingend gegeben.

Einen Ausweg aus diesem Dilemma bieten Alternativ-Lösungen. Das Computerprogramm berechnet im vorhinein mehrere gute alternative Lösungen und ein Experte entscheidet unter Zuhilfenahme aktueller oder zusätzlicher Informationen, welche er davon realisiert. Gute Aussichten bieten Alternativ-Lösungen, die mit einer **“Bestrafungs-Methode”** (*Penalty Method*) berechnet worden sind.

Kapitel 2 stellt zwei grundlegende Ansätze für Bestrafungs-Methoden vor, die (normale) *Penalty Method* und die *Mutual Penalty Method*. Die *Penalty Method* berechnet zu einer vorgegebenen optimalen Lösung eine alternative Lösung. Sie ist bereits aus Publikationen von Althöfer und Kollegen (siehe [ABS 2002], [ABSS 2004], [Ber 2000]) und vor allem aus der Dissertation von Schwarz (siehe [Sch 2003]) bekannt. Die der *Mutual Penalty Method* zugrunde liegende Idee wurde aus der ebenfalls von Schwarz formulierten *Linear Programming Penalty Method* entwickelt. Sie berechnet keine Alternativlösung zu einer vorgegebenen Referenz-Lösung, sondern direkt ein Lösungs-Paar.

Bei beiden Methoden erlaubt ein Strafparameter $\varepsilon > 0$, grundlegende Eigenschaften der zu berechnenden alternativen Lösungen zu steuern. Je größer der Strafparameter ε ist, desto weniger Teilkomponenten haben die Lösungen des Lösungs-Paares gemeinsam. Im Gegenzug nimmt die Güte der Alternativ-Lösungen im Sinne der Zielfunktion ab.

Die richtige Wahl des Strafparameters ε und der damit verbundene Vorteil, aus zwei Kandidaten-Lösungen aussuchen zu können, wird intensiv in Kapitel 3 mit Hilfe von Computer-Experimenten für verschiedene Optimierungs-Probleme (Kürzeste-Wege-Problem, Rundreiseproblem, Zuordnungsproblem) untersucht. Grundidee für die Experimente ist folgender 3-Phasen-Ablauf:

1. In einer ersten Phase werden zwei Lösungen berechnet (z.B. die optimale Lösung und die Alternativ-Lösung für einen gewählten Strafparameter ε).
2. Danach wird jede einzelne Teilkomponente des Problems gemäß folgendem $(p; 1, c)$-Modell, unabhängig von allen anderen Teilkomponenten, gestört: Mit Wahrscheinlichkeit p werden die Kosten der Komponente mit einem zufälligen Faktor λ_c multipliziert. Der Faktor λ_c wird dabei jeweils unabhängig und gleichverteilt aus dem Intervall $[1, c] \subset \mathbb{R}$ gewählt.
   Mit Wahrscheinlichkeit $(1 - p)$ bleibt die Komponente ungestört.
3. In der letzten Phase darf zwischen den beiden Lösungen gewählt werden.

   $\rightarrow$ Je kostengünstiger die bessere der beiden Lösungen hier im Durchschnitt ist, desto besser ist der gewählte Strafparameter ε.

Es werden alle in Kapitel 2 vorgestellten Varianten der Bestrafungs-Methoden in die Untersuchung einbezogen und miteinander verglichen. Es stellt sich z.B. heraus, dass

- die erzeugten Alternativ-Lösungen in allen Fällen mit wachsendem ε bis zu einem optimalen $\varepsilon_*(p, c)$ monoton besser werden und ab diesem dann wieder monoton schlechter.
- sich durch Alternativ-Lösungen, besonders im Fall von seltenen aber dafür heftigen Störungen, ein deutliches Verbesserungs-Potential ergibt.
- die *Mutual Penalty Method* zwar nicht schlechter, allerdings auch nicht besser abschneidet, als die normale *Penalty Method*. In der Praxis ist damit die normale *Penalty Method*, aufgrund ihrer einfacheren Implementierung und des geringeren Rechen-Aufwands, vorzuziehen.

In kompakter Darstellung werden in Kapitel 4 mehrere weitere Fragestellungen untersucht, z.B. wie sensibel die Ergebnisse von der Wahl des Strafparameters abhängen, inwieweit die Ergebnisse auf andere Störmodelle übertragbar sind, oder wie viel besser die Ergebnisse sein würden, wenn das mathematische Modell perfekt gelöst werden könnte. Außerdem wird eine Verbindung zwischen den Bestrafungs-Methoden und der Multi-Kriteriellen Optimierung hergestellt. Hier wird eine gemeinsame potentielle Schwäche aller vorgestellten Bestrafungs-Methoden deutlich, die in zukünftiger Forschung Ansatzpunkt für weitere Verbesserungen sein kann.

Kapitel 5 fasst die wichtigsten Ergebnisse aus der gesamten Arbeit zusammen und gibt eine Übersicht über mögliche zukünftige Untersuchungen.

Acknowledgements

I especially wish to thank my advisor Ingo Althöfer for his continual help and encouragement, as well as many invaluable discussions.

In addition, the following proofreaders helped eliminating many errors: Stefan Schwarz, Raymond Georg Snatzke and Lars Stoppe.

Finally, I thank my family and friends for their constant help, encouragement and patience with my frustration and mood swings during the last three years.

This thesis was made possible by a doctoral grant from the Research Training Group "Approximation and Algorithms" (Graduiertenkolleg "Approximation und algorithmische Verfahren"), supported by the German Research Foundation (Deutsche Forschungsgemeinschaft) and the State of Thuringia (Freistaat Thüringen).

Chapter 1

Introduction

Assume we are given an optimization problem that, in abstract form, represents a *"real world"* problem. During the transfer process from the real world problem to the mathematical model, often compromises have to be made that result in a mathematical model that does not represent the real world problem perfectly. Then the optimal solution for the mathematical model can differ from the optimal solution for the real world problem. Reasons for a such compromises can be:

(i) **Unstable data**: stochastic variation of the problem data *(for instance, travel times in a road network)*

(ii) data or influences are ignored in the mathematical model because

- **Lack of Knowledge**: data retrieval problems
  (political circumstances, cyclic or acyclic fluctuations, calendar, weather, special events)
- **Insufficient calculating power/ability**: complexity
  (no algorithm available that can handle the complexity, time or memory restrictions, ...)

(iii) **Unclear goals**: different users have different utility functions in mind
(short travel time versus comfort versus price versus ...)

There are several approaches that try to handle unstable data. *Robust Optimization* searches for solutions for the mathematical model that are as insensitive as possible to changes. Then small perturbations in the problem data do not result in a large drop of the objective function. *Stochastic Programming* builds a model of the stochastic variation of the problem data and searches for the solution that is best in the average case.

Another possibility to handle different types of imperfect models is the generation of alternative solutions. Here the problem solver generates not only the apparently best solution but also some different candidate solutions. Then the planner (human expert) chooses – with the help of additional expert knowledge – his favorite solution, which finally gets realized.

Penalty methods have a good chance to find reasonable alternatives: For instance, the best solution is computed, and then certain building blocks of this solution are penalized. The best solution with respect to this modification represents an alternative solution.

S. Schwarz was first to study this topic for the shortest path problem in his doctoral dissertation (see [Sch 2003]). We present a more complex simulation of uncertain data, give results for shortest paths and additional optimization problems (assignment problem, traveling salesman problem), and provide a widened experimental analysis.

1.1 Optimization Problems under Investigation

1.1.1 Sum Type Optimization Problems

The following definition is valid for all optimization problems investigated in this thesis.

Definition 1.1.1 (Sum Type Optimization Problem)
Consider a triple $P = (E, F, w)$. *Let* E *be an arbitrary countable set and* F *a subset of the power set* $\mathcal{P}(E)$ *of* E. *We call* E *the base set and the elements of* F *feasible subsets of* E. *Let* $w : E \to \mathbb{R}$ *be a real-valued weight function on* E. *For every* $S \in F$ *we set* $w(S) = \sum_{e \in S} w(e)$. *The optimization problem* $\min_{S \in F} w(S)$ *is called a Sum Type Optimization Problem.*

Remarks:

- We use the abbreviation "$\sum$-type problem" instead of "Sum Type Optimization Problem".
- The elements $S \in F$ are called feasible solutions.
- The definition of a $\sum$-type problem covers not only minimization problems. By substitution of w by $-w$, a maximization problem can be expressed as a minimization problem.

In the next subsection we briefly list five well known $\sum$-type optimization problems. The first three problems (Shortest Path Problem, Assignment Problem, Traveling Salesman Problem) are of special interest. We use them for experiments later in this thesis. The remaining two problems (Knapsack Problem, Sequence Alignment) are only listed as further examples for $\sum$-type problems.

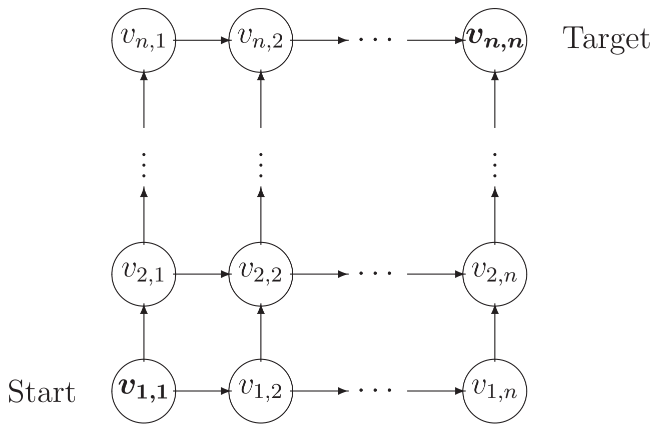
1.1.2 Examples of Sum Type Optimization Problems

1.1.2.1 The Shortest Path Problem

Consider a directed graph $G = (V, E)$ and a function $w : E \to \mathbb{R}^+$, assigning a length to every edge of the graph. Let s and t be two distinguished nodes of G. The task is to find a shortest path from s to t. The length of a path is the sum of the lengths of the edges used.

Here we deal with shortest paths in weighted directed grid graphs only.

Definition 1.1.2 (Weighted Directed Grid Graph)
A weighted directed grid graph of size $n \times n$ *is a graph* $G = (V, E, w)$ *with* $V = \{v_{i,j} : 1 \leq i, j \leq n\}$ *and* $E = \{(v_{i,j}, v_{k,l}) : i, j, k, l \in \{1, \ldots, n\}$ *with* $0 \leq (k - i) \leq 1$, $0 \leq (l - j) \leq 1$ *and* $(k - i) + (l - j) = 1\}$ *and a weight function* $w : E \to \mathbb{R}$. *Starting node* s *is the lower left node* $v_{1,1}$ *and target node* t *the upper right node* $v_{n,n}$



The representation as a $\sum$-type problem is obvious. The set of edges E is the base set, the family of all edge sets that represent simple paths from s to t is the set of feasible subsets F. Then we have to minimize $w(S)$ over all $S \in F$.

1.1.2.2 The Assignment Problem

Consider a set of n workers $W_1, W_2, \ldots, W_n$, a set of n jobs $J_1, J_2 \ldots, J_n$ and a cost function $c : W \times J \to \mathbb{R}$. The task is to find for each worker exactly one job such that all jobs are done with minimal total cost.

Here the base set E is the cross product $W \times J$ and F is the family of all subsets of E which represent a complete matching between W and J. Thus, we obtain a representation as a $\sum$-type problem: minimize $c(S)$ over all $S \in F$.

1.1.2.3 The Traveling Salesman Problem

Consider a set of n cities $C = \{c_1, c_2, \ldots, c_n\}$ and for each pair $\{c_i, c_j\}$ of distinct cities a symmetric distance $d(c_i, c_j) = d(c_j, c_i)$. The task is to find a permutation π of the cities that minimizes the tour length:

$$l(\pi) := \sum_{i=1}^{n-1} d\big(c_{\pi(i)}, c_{\pi(i+1)}\big) + d\big(c_{\pi(n)}, c_{\pi(1)}\big).$$

Here the base set E is the cross product $C \times C$ and F is the family of all permutations of the cities which represent a closed tour through all cities. This way we get a representation as a $\sum$-type problem : minimize $l(\pi)$ over all $\pi \in F$.

1.1.2.4 The Knapsack Problem

Consider a set of items $I = \{I_1, I_2, \ldots, I_n\}$, a weight function $w : I \to \mathbb{R}^+$, a value function $v : I \to \mathbb{R}^+$, and a knapsack capacity C.

The task is to find the most valuable set of items whose sum of weights does not exceed the capacity of the knapsack.

Choosing I as base set and F as the family of all subsets whose sum of weights is smaller or equal to C, we get a representation as a $\sum$-type problem: maximize $v(S)$ over all $S \in F$.

1.1.2.5 The Sequence Alignment Problem

Consider two strings $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_m)$ of characters of an alphabet Γ and a scoring function $v : \Gamma \cup \{\_\} \times \Gamma \cup \{\_\} \to \mathbb{R}$. In an alignment of A and B it is allowed to insert gaps ("_") in both strings, resulting in new strings $\widetilde{A}$ and $\widetilde{B}$. $\widetilde{A}$ and $\widetilde{B}$ have the same length $l(\widetilde{A}, \widetilde{B})$.

The task is to find the alignment with the highest total score $\displaystyle\sum_{i=1}^{l(\widetilde{A},\widetilde{B})} v(\widetilde{a}_i, \widetilde{b}_i)$.

As base set we can use $\Gamma \cup \{\_\} \times \Gamma \cup \{\_\}$. The family F consists of all subsets S which represent an alignment of A and B. Thus we have our $\sum$-type problem: maximize $v(S)$ over all $S \in F$.

1.2 Overview

This thesis is organized as follows. Chapter 2 introduces two basic concepts of penalty methods in detail. First, the (normal) *penalty method* generates an appropriate alternative solution for a given reference solution. Second, the *mutual penalty method* directly generates two good candidate solutions. In both cases a penalty parameter allows to control fundamental properties of the solutions. These methods will be demonstrated both with exact algorithms and with heuristics.

Chapter 3 is the heart of the thesis. It starts with an introduction to the experimental setup and the simulation of uncertain data.

> In a planning phase we are given a $\sum$-type problem $P = \{E, F, w\}$. We are allowed to prepare two different solutions: a first alternative S_{a_1} and a second alternative S_{a_2}. Then – in the time between the planning phase and the execution phase – the edge weights change in some random way from w to $\widehat{w}$. In the execution phase we choose the candidate solution of the pair (S_{a_1}, S_{a_2}) that is better with respect to the new weights $\widehat{w}$.

Our simulation of uncertain data is based on a perturbation model introduced by Schwarz (see [Sch 2003]). We extend this model in such a way that we can control not only the expected size of perturbations but also the frequency.

Sections 3.3 and 3.4 analyze the improvement achieved by the choice from two candidate solutions and the optimal choice of the penalty parameter in contrast to the situation where only one solution is available. Section 3.3 deals with exact algorithms and Section 3.4 with heuristics. Section 3.5 comments on the stability and variation of the simulation results.

Chapter 4 concisely discusses six additional questions that arouse in Chapter 3 or that are not taken into account there. Chapter 5 presents conclusions and open problems.

Chapter 2

Penalty Methods

2.1 Introduction

There are two main criteria for *good* alternatives. First, an alternative should be *good* with respect to the objective function. Second, an alternative should have *little in common* with the original solution. Otherwise it would not be a true alternative. Taking only the first criterion into account, one may simply compute k-best solutions. Practical experiences show that this method generally does not fulfill the second criterion. Typically, k-best solutions are only *micro mutations* of the original optimum, differing only in very small details.

As an example, Figure 2.1 shows the 10^5 shortest paths of a 100×100 grid graph with random edge lengths ($w(e)$ uniformly distributed in $[0, 1] \subset \mathbb{R}$, independently and identical distributed for all $e \in E$). Figure 2.2 shows only the shortest path and the 10^5-th shortest path of the same graph.

To find truly alternative solutions with the k-best method one often has to compute k-best solutions for very large values of k [Ber 2000]. But even a very large k does not guarantee the diversity of the alternative and the original optimal solution. A larger k gives a better chance to find a true alternative but not necessarily a good chance. This can be seen in Figure 2.3. Every point in the diagram represents one of the 10^5 shortest paths from the example in Figure 2.1. Horizontally the solutions are ordered by their length and vertically they are ordered by the length of the way they share with the shortest path. Even among the worst solutions (within this set) there are some that are nearly identical to the shortest path.

A method that allows to directly control the diversity is the Penalty Method (see [Ber 2000], [ABS 2002], [Sch 2003]).

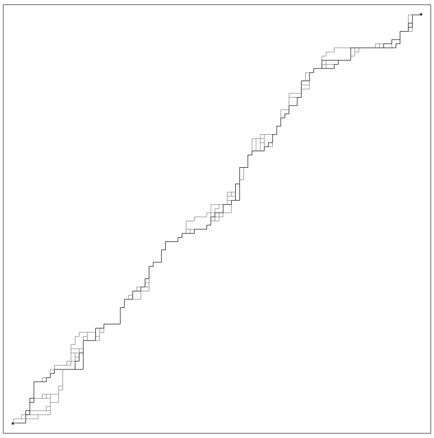


Figure 2.1: The 10^5 shortest paths of a 100×100 grid graph, printed in overlap.

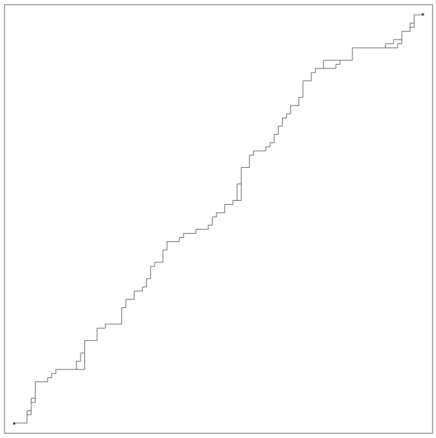


Figure 2.2: The shortest and the 10^5-th shortest path of a 100×100 grid graph, printed in overlap.

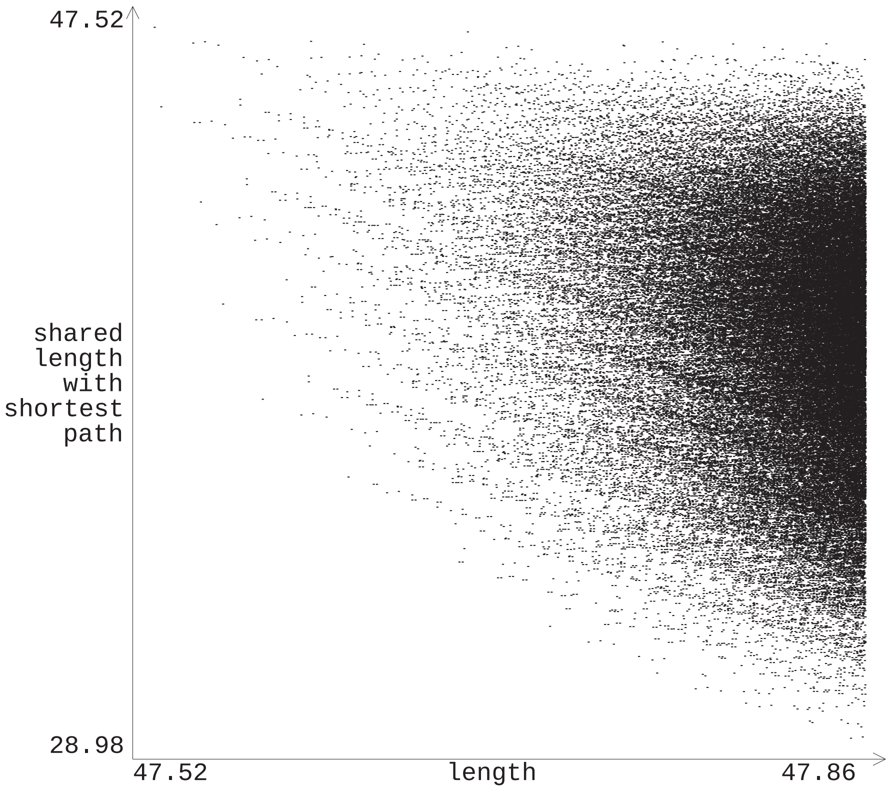


Figure 2.3: The 10^5 shortest paths of a 100×100 grid graph.

2.2 Generating Alternative Solutions by Penalties

2.2.1 The Penalty Method

The following definitions can also be found in [ABS 2002] (pp. 2-3) and [Sch 2003] (pp. 7-9), although with different names and symbols.

Definition 2.2.1 (Penalty Method, [ABS 2002])
Consider a $\sum$-type problem $P = (E, F, w)$ with base set E, set of feasible subsets F and real-valued weight function w. We call S_0 the optimal solution of the problem $\min_{S\in F} w(S)$. If there is more than one optimal solution, we may choose any of them.
For every $\varepsilon > 0$, let S_ε be one of the optimal solutions of the problem

$$\min_{S\in F} \widetilde{w}(S)$$
$$\text{with} \quad \widetilde{w}(e) := \begin{cases} (1+\varepsilon)\cdot w(e) & : \text{ if } e \in S_0 \\ w(e) & : \text{ otherwise.} \end{cases}$$

We call S_ε an ε-penalty solution *and the approach* Penalty Method.

Additionally, we define the solution S_∞ of the problem

$$S_\infty := \operatorname{lex}\min_{S\in F}(w(S\cap S_0), w(S)).$$

'Lex min' means lexicographic minimization: S_∞ has minimal weight intersection with S_0 and among all such solutions S_∞ is one with minimal weight $w(S)$.

In general the solutions S_ε can be found with the same algorithm which solves the unpunished problem $\min_{S\in F} w(S)$. We only have to change the weight values w for the elements $e \in S_0$. So, computing the alternative is not harder than computing the optimal solution.

Depending on the problem structure it can be advantageous to penalize not only the shared parts $(S \cap S_0)$ but also solution parts that are "close" to the original solution S_0. A generalized version of the penalty method gives much room for the choice of a suitable type of punishment.

Definition 2.2.2 (Generalized Penalty Method, [ABS 2002])
Consider a $\sum$-type problem $P = (E, F, w)$ with base set E, set of feasible subsets F, weight function $w : E \to \mathbb{R}$ and positive function $p : E \to \mathbb{R}^+$.

For every $\varepsilon > 0$, let S_ε be one of the optimal solutions of the problem

$$\min_{S \in F} \widetilde{w}(S)$$
$$\text{with} \quad \widetilde{w}(S) := w(S) + \varepsilon \cdot p(S).$$

We call S_ε an ε-penalty solution.

Additionally we define the solution S_∞ of the problem

$$S_\infty := \operatorname{lex}\min_{S \in F}(p(S), w(S)).$$

'Lex min' means lexicographic minimization: S_∞ has minimal value $p(S)$ and among all such solutions minimal value $w(S)$.

2.2.2 Theoretical Background

The penalty method has nice properties that make it a useful approach to find alternative solutions. These properties are given by the following theorem:

Theorem 2.2.3 (Althöfer, Berger, Schwarz [ABS 2002])
Let $w : E \to \mathbb{R}$ be a real-valued function and $p : E \to \mathbb{R}^+$ be a positive real-valued function on E. Let S_ε be defined according to Definition 2.2.2. The following four statements hold:

1. *$p(S_\varepsilon)$ is weakly monotonically decreasing in ε.*
2. *$w(S_\varepsilon)$ is weakly monotonically increasing in ε.*
3. *$w(S_\varepsilon) - p(S_\varepsilon)$ is weakly monotonically increasing in ε.*
4. *$w(S_\varepsilon) + \varepsilon \cdot p(S_\varepsilon)$ is weakly monotonically increasing in ε.*

So the larger the penalty parameter ε, the less punished parts are contained in the solution S_ε. For the basic penalty method this means that alternative solution S_ε and original optimum S_0 have less and less in common. In compensation, the original objective value $w(S)$ is getting worse.

The tradeoff between these two criteria (objective value vs. diversity) is of special interest for this thesis. In Chapter 3 we describe and discuss several experiments to determine a penalty parameter ε that best balances this tradeoff.

2.2.3 The Penalty Method with Heuristics

Assume we are given a problem which we are not able to solve exactly within a given time. Instead, we might be able to solve the problem heuristically. Then we are also able to calculate heuristically an alternative solution using the penalty approach. We only have to replace *"optimal solution"* by *"heuristic solution"* in Definitions 2.2.1 or 2.2.2. The basic case is given in Definition 2.2.4.

Definition 2.2.4 (Penalty Method with Heuristics)
Consider a $\sum$-type problem $P = (E, F, w)$ with base set E, set of feasible subsets F and real-valued weight function w. Let $\check{S}_0$ be a heuristic solution of the problem $\min_{S \in F} w(S)$.
For every $0 < \varepsilon < \infty$, let $\check{S}_\varepsilon$ be a heuristic solution of the problem

$$\min_{S \in F} \widetilde{w}(S)$$
$$\text{with} \quad \widetilde{w}(e) := \begin{cases} (1+\varepsilon) \cdot w(e) & : \text{ if } e \in \check{S}_0 \\ w(e) & : \text{ otherwise.} \end{cases}$$

We call $\check{S}_\varepsilon$ a heuristic ε-penalty solution *and the approach* Penalty Method with Heuristics.

Using the penalty method with a heuristic, we are not able to give a theorem that guarantees monotonicity. For the exact case we know that the value $w(S_\varepsilon)$ of an ε-penalty solution S_ε is getting worse for increasing ε. A heuristic generally finds a solution that is not globally optimal. Therefore, it produces 'good' solutions for some penalty parameters and 'bad' solutions for other penalty parameters as Example 2.2.5 shows. But if we generate many instances of the problem and calculate average values the penalty method works smooth again as Example 2.2.6 shows. So Schwarz conjectured that for many problems and heuristics the penalty method with heuristics behaves on average as the penalty method with exact algorithms (see [Sch 2003]).

Under the assumption that this conjecture is true we take a look at these (difficult) problem classes we solve heuristically with local search algorithms (see Appendix A.2 for more details): the traveling salesman problem and the assignment problem. Our aim is to check whether the observations we made for the exact case remain true (on average) for the penalty method with heuristics.

Example 2.2.5
Consider the traveling salesman problem (TSP) as defined in Subsection 1.1.2.3. We generated one instance of a TSP by randomly choosing n cities, uniformly distributed in the unit square $[0,1]^2$ and afterwards calculating the symmetric euclidian distances $d(c_i, c_j)$ $(i, j = 1, \ldots, n,\ i \neq j)$ between all cities. With a local search algorithm (as described in Appendix A.2) we calculated a locally

optimal solution $\check{S}_0$ and a whole set of heuristic ε-penalty solutions $\check{S}_\varepsilon$ for all $\varepsilon \in \{0.025, 0.05, \ldots, 2.00\}$. Figure 2.4 shows the objective values $w(\check{S}_\varepsilon)$ (marked by $\square$) and the lengths of the shared edges $w(\check{S}_0 \cap \check{S}_\varepsilon)$ (marked by $\times$) of the heuristic ε-penalty solutions with respect to the penalty parameter ε.

Example 2.2.6
We generated 10^6 instances of the TSP in the same way as in Example 2.2.5. For each instance we calculated a locally optimal solution $\check{S}_0$ and heuristic ε-penalty solutions $\check{S}_\varepsilon$ for all $\varepsilon \in \{0.025, 0.05, \ldots, 2.00\}$. From the data of the 10^6 instances we finally calculated average lengths $\bar{w}(\check{S}_\varepsilon)$ (marked by $\square$) and the average lengths of the shared edges $\bar{w}(\check{S}_0 \cap \check{S}_\varepsilon)$ (marked by $\times$) for all penalty parameters ε. The results are shown in Figure 2.5.

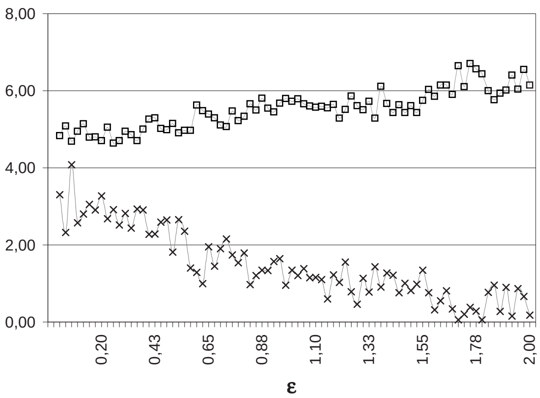


Figure 2.4: Lengths $w(\check{S}_\varepsilon)$ (marked by $\square$) and lengths of the shared edges $w(\check{S}_0 \cap \check{S}_\varepsilon)$ (marked by $\times$) of heuristic solutions of a randomly generated TSP with $n = 25$ cities.

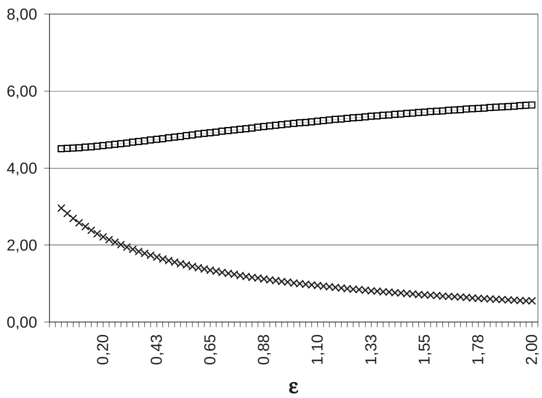


Figure 2.5: Average lengths $\bar{w}(\check{S}_\varepsilon)$ (marked by $\square$) and average lengths of the shared edges $\bar{w}(\check{S}_0 \cap \check{S}_\varepsilon)$ (marked by $\times$) of heuristic solutions of 10^6 randomly generated TSPs with $n = 25$ cities.

2.3 Generating Solution Pairs by Mutual Penalties

When we were looking for a good solution pair in Section 2.2 we always generated an optimal solution in a first step. Afterwards – in a second step – we looked for a suitable alternative solution. Although this approach is computationally very easy and seems to be quite successful, it is in no way clear that the best single solution must be part of the pair of solutions.

In this section we follow a different approach. From the outset we look for a good pair of solutions.

In an ultimate sense the best pair would be two completely different, globally optimal solutions. In general such a pair does not exist. We try to come as close as possible to this case by introducing mutual penalties for shared parts of the solutions.

2.3.1 The Mutual Penalty Method

The idea for this method is taken from the doctoral dissertation of S. Schwarz (see [Sch 2003] p.20). There he introduced a "Linear Programming – Penalty Method" that is a special case of our method. His method focuses on generating good pairs of short paths in linear programming. But this idea allows a much more general approach that does not depend on linear programming.

Definition 2.3.1 (Mutual Penalty Method (MPM))
Consider a $\sum$-type problem $P = (E, F, w)$ with base set E, set of feasible subsets F and real-valued weight function w.
For every $\varepsilon > 0$, let $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ be one of the optimal pairs of solutions of the penalized problem

$$\min_{\{S_1,S_2\}\in F} \widetilde{w}_\varepsilon(S_1, S_2)$$

with

$$\widetilde{w}_\varepsilon(S_1, S_2) := \sum_{e\in S_1} w(e) + \sum_{e\in S_2} w(e) + \varepsilon \cdot \sum_{e\in S_1\cap S_2} w(e)$$

We call $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ an ε-penalty pair *and the approach* Mutual Penalty Method.
*We call the task to solve a $\sum$-type problem by the Mutual Penalty Method MPM-*problem.

Additionally we define the solution pair $\{S_{1(\infty)}, S_{2(\infty)}\}$ of the problem

$$\operatorname{lex} \min_{\{S_1,S_2\}\in F} (w(S_1 \cap S_2), w(S_1) + w(S_2)).$$

'Lex min' means lexicographic minimization: the solutions $\{S_{1(\infty)}, S_{2(\infty)}\}$ have minimal weight intersection with each other and among all such solutions $\{S_{1(\infty)}, S_{2(\infty)}\}$ have minimal sum of weights $(w(S_1) + w(S_2))$.

Assume we are given a $\sum$-type problem $\min_{S\in F} w(S)$ with $w(S) = \sum_{e\in S} w(e)$ (see 1.1.1). For every $e \in E$ we know its weight $w(e)$ that we can interpret as cost. So, if a solution uses a certain element $e_1 \in E$, it costs $w(e_1)$. The general idea is to penalize when both solutions of the pair use the same element e. So we assume that every element $e \in E$ is available only one time for cost $w(e)$. If the other solution requests the same element we have to pay some extra amount. This is accomplished by a penalty parameter $\varepsilon > 0$. Thus the total cost to use an element e twice is $(2+\varepsilon)\cdot w(e)$.

Although we restrict our considerations to this approach it can be defined in a more general way as follows:

Definition 2.3.2 (Generalized Mutual Penalty Method ($GMPM$))
Consider a $\sum$-type problem $P = (E, F, (w_1, w_2))$ with base set E, the set of feasible subsets F, two real-valued weight functions $w_1, w_2 : E \to \mathbb{R}$ and a positive real-valued penalty function $p : E \to \mathbb{R}^+$.
For every $\varepsilon > 0$, let $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ be one of the optimal pairs of solutions of the penalized problem

$$\min_{\{S_1,S_2\}\in F} \widetilde{w}_\varepsilon(S_1, S_2)$$

$$\textit{with} \qquad \widetilde{w}_\varepsilon(S_1, S_2) := w_1(S_1) + w_2(S_2) + \varepsilon \cdot p(S_1, S_2)$$

We call $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ an ε-penalty pair with respect to (w_1, w_2, p).

Additionally we define the solution pair $\{S_{1(\infty)}, S_{2(\infty)}\}$ of the problem

$$\operatorname{lex} \min_{\{S_1,S_2\}\in F} (p(S_1, S_2), w_1(S_1) + w_2(S_2)).$$

'Lex min' means lexicographic minimization: the solutions $\{S_{1(\infty)}, S_{2(\infty)}\}$ have minimal value $p(S_1, S_2)$ and among all such pairs minimal value $(w_1(S_1)+w_2(S_2))$.

The $GMPM$ is equivalent to the basic MPM if $w(e) := w_1(e) = w_2(e)$ for all $e \in E$ and the penalty function $p : E \to \mathbb{R}$ is chosen as

$$p(S_1, S_2) := \sum_{e\in\{S_1\cup S_2\}} \begin{cases} 0 & : \quad e \notin \{S_1 \cap S_2\}, \\ w(e) & : \quad e \in \{S_1 \cap S_2\}\,. \end{cases}$$

In general, the (normal) penalty method and the mutual penalty method lead to different results. The normal penalty method calculates the optimal solution and a suitable alternative. The mutual penalty method generates a pair of good solutions with relatively small overlap. The normal optimal solution is not necessarily part of this pair.

Figure 2.6 shows an example with short paths. The shortest path from s to t is the path $P_0 = (a, b, c)$ with length 30. For all $\varepsilon > 0.1$ the ε-penalty pair contains the path P_0 and the path $P_1 = (d, c)$. The pair (P_0, P_1) has a total length of 62 and shared length of 10. For all $\varepsilon > 0.4$ the mutual penalty method produces the pair (P_1, P_2) with $P_2 = (a, e)$. This pair has a total length of 66 and a shared length of zero. The solution pair does not contain the shortest path P_0.

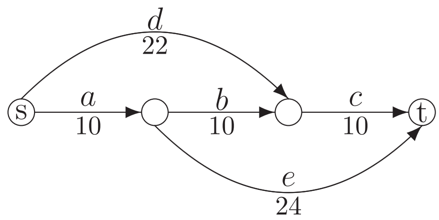


Figure 2.6: Shortest Path Example – The ε-penalty pair $\{(d,c),(a,e)\}$ for $\varepsilon > 0.4$ does not contain the single optimal path (a,b,c).

2.3.2 Theoretical Background

For the penalty method Theorem 2.2.3 gives fundamental results how the generated alternatives depend on the penalty parameter ε. Theorem 2.3.3 gives analogous results for the generalized mutual penalty method.

Theorem 2.3.3
Consider a $\sum$-type problem $P = (E, F, (w_1, w_2))$ with base set E, set of feasible subsets F, two real-valued weight functions $w_1, w_2 : E \to \mathbb{R}$ and positive real-valued penalty function $p : E \to \mathbb{R}^+$.
Let $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ be defined for $\varepsilon \in \mathbb{R}^+$ according to Definition 2.3.2. It holds:

(i) $p(S_{1(\varepsilon)}, S_{2(\varepsilon)})$ *is weakly monotonically decreasing in ε.*

(ii) $w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)})$ *is weakly monotonically increasing in ε.*

(iii) $w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) - p(S_{1(\varepsilon)}, S_{2(\varepsilon)})$ *is weakly monotonically increasing in ε.*

(iv) $w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \varepsilon \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)})$ *is weakly monotonically increasing in ε.*

Proof
Let ε and δ be two arbitrary positive real numbers with $0 \le \delta < \varepsilon$. Because of the definition of $(S_{1(\varepsilon)}, S_{2(\varepsilon)})$ and $(S_{1(\delta)}, S_{2(\delta)})$ the following inequalities hold.

(i) If $\varepsilon < \infty$ we have

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \varepsilon \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \le w_1(S_{1(\delta)}) + w_2(S_{2(\delta)}) + \varepsilon \cdot p(S_{1(\delta)}, S_{2(\delta)}), \tag{2.1}$$

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \delta \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \ge w_1(S_{1(\delta)}) + w_2(S_{2(\delta)}) + \delta \cdot p(S_{1(\delta)}, S_{2(\delta)}). \tag{2.2}$$

Subtracting (2.2) from (2.1) we get

$$\begin{aligned} & (\varepsilon - \delta) \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \le (\varepsilon - \delta) \cdot p(S_{1(\delta)}, S_{2(\delta)}) \qquad | : (\varepsilon - \delta) \\ \Leftrightarrow \quad & p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \le p(S_{1(\delta)}, S_{2(\delta)}). \end{aligned} \tag{2.3}$$

In case of $\varepsilon = \infty$ inequality (2.3) follows directly from the definition of $\{S_{1(\infty)}, S_{2(\infty)}\}$.

(ii) Subtracting (2.3) multiplied with δ from (2.2) we get

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) \geq w_1(S_{1(\delta)}) + w_2(S_{2(\delta)}). \tag{2.4}$$

(iii) Subtracting (2.3) from (2.4) we get

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) - p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \geq w_1(S_{1(\delta)}) + w_2(S_{2(\delta)}) - p(S_{1(\delta)}, S_{2(\delta)}).$$

(iv) With $\varepsilon > \delta \geq 0$ we have

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \varepsilon \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \geq w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \delta \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)})$$

Together with (2.2) we get

$$w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}) + \varepsilon \cdot p(S_{1(\varepsilon)}, S_{2(\varepsilon)}) \geq w_1(S_{1(\delta)}) + w_2(S_{2(\delta)}) + \delta \cdot p(S_{1(\delta)}, S_{2(\delta)})$$

□

Conclusion:
For the generalized mutual penalty method we proved analogous results as are known for the (normal) penalty method. So this method can also be used to control the main quality criteria of alternative solutions. With increasing penalty parameter ε we get solution pairs $(S_{1(\varepsilon)}, S_{2(\varepsilon)})$ which contain less and less penalized (= shared) parts $p(S_{1(\varepsilon)}, S_{2(\varepsilon)})$. In compensation, the sum of the objective values $(w_1(S_{1(\varepsilon)}) + w_2(S_{2(\varepsilon)}))$ gets worse.

2.3.3 Solving an MPM-Problem

In contrast to the normal penalty method we can not find the ε-penalty pair $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ with the same algorithm that solves the original problem.

Consider a $\sum$-type problem $P = (E, F, w)$. Every feasible solution $S \in F$ is an extremal point of the power set $\mathcal{P}(E) = \{0,1\}^E$ of the base set E. The extremal points of the convex hull $conv(F) \subseteq [0,1]^E$ are the feasible solutions itself and the objective function is linear. Thus there is an optimal solution that is an extremal point of $conv(F)$. This leads directly to

Remark 2.3.4
Every $\sum$-type problem $P = (E, F, w)$ can be represented by a linear program LP.

The corresponding MPM-problem can be written as another $\sum$-type problem $(E_{MPM}, F_{MPM}, \widetilde{w})$ with base set $E_{MPM} := E \times E$, set of feasible subsets $F_{MPM} := F \times F$ and real-valued weight function $\widetilde{w}(e) : E_{MPM} \to \mathbb{R}$ as defined in Definition 2.3.1 or 2.3.2.

Remark 2.3.5
Every MPM-problem is a $\sum$-type problem itself and thus can be represented by a linear program LP.

The LP representation above is quite complicated and mostly of theoretical interest. For the shortest path problem Schwarz showed how to model the corresponding linear program - that solves the MPM-problem - much easier (see [Sch 2003] p.21). For the assignment problem we show in Subsection 2.3.3.1 how to model the penalized problem as a linear program. This example illustrates how the linear programming approach may work for other problems.

For certain $\sum$-type problems an exponential number of constraints is needed to represent the polytope of the feasible solutions in a linear program. Such problems are not efficiently solvable by linear programming. Certain $\sum$-type problems are much easier formulated as integer linear programs (ILPs). But integer linear programming is NP-complete in general, thus ILPs are hard to solve, too.

With respect to possible restrictions concerning the computation time it might be better to solve the MPM-problem approximately with a heuristic algorithm. For example, it is quite easy to find locally optimal solutions of the MPM-problem with local search algorithms. This approach is shown in Subsection 2.3.3.2.

2.3.3.1 Linear Programming

The Assignment Problem

The assignment problem presented in Subsection 1.1.2.2 can be stated as a linear program LP^{asp}:

$$\begin{aligned}
\min & \sum_{i,j=1,\dots,n} c_{ij} \cdot x_{ij} \\
\text{s.t.:} & \sum_{i=1,\dots,n} x_{ij} = 1 \quad \text{for all } j = 1, \dots, n, \\
& \sum_{j=1,\dots,n} x_{ij} = 1 \quad \text{for all } i = 1, \dots, n, \\
& 0 \le x_{ij} \le 1 \text{ for all } i, j = 1, \dots, n.
\end{aligned}$$

A binary variable $x_{i,j} \in \{0, 1\}$ indicates an assignment of worker W_i to job J_j and $c_{i,j}$ corresponds to the cost of the assignment.

A feasible solution of the linear program is not necessarily a regular assignment, as the variables x_{ij} might have non-integer values. The assignment problem is a special case of the minimum cost flow problem. There is a theorem for network flow problems ([AMO 1993], p.318) stating that "*if all arc capacities and supplies/demands of nodes are integer, the minimum cost flow problem always has an optimal solution where all variables are integer*". Thus the existence of an optimal solution where all variables are integers is guaranteed.

This integer solution is easy to find. If one is using a simplex algorithm to solve the linear program, then the solution will always consist only of integers since every extremal point consists only of integers. In contrast, an interior point method does not necessarily find an integer solution.

For the corresponding MPM-problem we double the number of variables. Every assignment of a worker W_i to a job J_j gets two variables x_{ij} and y_{ij}. If an assignment $W_i \to J_j$ is used in both solutions, then both variables x_{ij} and y_{ij} have value 1. If an assignment $W_i \to J_j$ is used only in one solution, then x_{ij} shall be 1 and y_{ij} shall be 0. Otherwise both variables x_{ij} and y_{ij} shall be 0. The linear program LP_ε^{asp} is given by

$$\min \sum_{i,j=1,\ldots,n} c_{ij} \cdot x_{ij} + (1+\varepsilon) \cdot c_{ij} \cdot y_{ij} \tag{2.5}$$

$$\text{s.t.:} \sum_{i=1,\ldots,n} x_{ij} + y_{ij} = 2 \quad \text{for all } j = 1, \ldots, n, \tag{2.6}$$

$$\sum_{j=1,\ldots,n} x_{ij} + y_{ij} = 2 \quad \text{for all } i = 1, \ldots, n, \tag{2.7}$$

$$0 \leq x_{ij}, y_{ij} \leq 1 \text{ for all } i, j = 1, \ldots, n.$$

A solution of LP_ε^{asp} does not give two simple assignments directly. The connection between the solution of the linear program LP_ε^{asp} and the original task to find two good assignments is established by Theorem 2.3.6. In [Sch 2003] an analogous theorem for short paths had been given.

Theorem 2.3.6
If the linear program LP_ε^{asp} has an optimal solution then it also has an optimal solution (x, y) with the following properties. There are disjoint sub-assignments $E_1, E_2, E_3 \subseteq E$ with

(i) $E_1 \cap E_2 = \emptyset$, $E_1 \cap E_3 = \emptyset$ *and* $E_2 \cap E_3 = \emptyset$,

(ii) $x_{ij} = 1$ *and* $y_{ij} = 0$ *for all pairs* $(W_i, J_j) \in E_1 \cup E_2$,

(iii) $x_{ij} = 1$ *and* $y_{ij} = 1$ *for all pairs* $(W_i, J_j) \in E_3$,

(iv) $x_{ij} = 0$ *and* $y_{ij} = 0$ *for all pairs* $(W_i, J_j) \notin E_1 \cup E_2 \cup E_3$,

(v) *$E_1 \cup E_3$ represents a complete assignment S_1 of jobs to workers and $E_2 \cup E_3$ represents a second complete assignment S_2 of jobs to workers. E_3 is the set of assignments used by both S_1 and S_2.*

(vi) *No other pair (Q_1, Q_2) of assignments is better than (S_1, S_2), i.e.,*

$$w(S_1) + w(S_2) + \varepsilon \cdot w(S_1 \cap S_2) \leq w(Q_1) + w(Q_2) + \varepsilon \cdot w(Q_1 \cap Q_2),$$

for all pairs (Q_1, Q_2)

That means the sum of the assignment costs of S_1 and S_2 plus a penalty for assignments used twice is minimal.

Proof:
The linear program LP_ε^{asp} is a special case of the minimum cost flow problem. Thus the existence of an optimal integer solution (x, y) with

$$x_{ij}, y_{ij} \in \{0, 1\} \tag{2.8}$$

can be derived from the theory of network flows ([AMO 1993], p.318).

From the objective function (2.5) and the integer property (2.8) we know that

$$(y_{ij} = 1) \Rightarrow (x_{ij} = 1). \tag{2.9}$$

As stated in *(iii)*, let E_3 be the set of assignments (W_i, J_j) with $x_{ij} = y_{ij} = 1$.

Consider (i_0, j_0) with $(W_{i_0}, J_{j_0}) \in E_3$. From (2.6) and (2.7) it follows that

$$x_{i_0 j} = y_{i_0 j} = x_{i j_0} = y_{i j_0} = 0 \quad \text{for all} \quad j \neq j_0 \quad \text{and all} \quad i \neq i_0 \tag{2.10}$$

Thus E_3 has to be the sub-assignment that both complete assignments S_1 and S_2 have in common.

Now we can remove all workers W_i and jobs J_j that are already part of the sub-assignment E_3. Let $|E_3|$ denote the number of assignments in E_3. From the Constraints (2.6) and (2.7) it follows that there remain $2(n - |E_3|)$ assignments $(W_i, J_j) \in E_1 \cup E_2$ with $x_{ij} = 1$ and $y_{ij} = 0$.

The following construction with alternatingly bi-colored edges proves that the remaining $2(n - |E_3|)$ single assignments can be decomposed into two complete assignments E_1 and E_2 of the remaining workers to the remaining jobs.

Without loss of generality, let $W_1, W_2, \ldots, W_{(n-|E_3|)}$ be the remaining workers and $J_1, J_2, \ldots, J_{(n-|E_3|)}$ the remaining jobs, respectively. We start with worker W_1 and the color red. (It might be advantageous to have the visualization of a bipartite graph in mind, as seen in Figures 2.7 and 2.8 below.) First, we color the edge that connects W_1 with it's lexicographically first assigned job J_{j_1}. We switch to color green. From Constraint (2.7) it follows that there is another worker W_{i_1} assigned to this job J_{j_1}. The edge that connects W_{i_1} and J_{j_1} we color green. From (2.6) we know that there is another job J_{j_2} assigned to worker W_{i_1}. If the connecting edge is still uncolored we color it with red. Now we color the other edge from J_{j_2} to a worker W_{i_2} again with green, and so on.

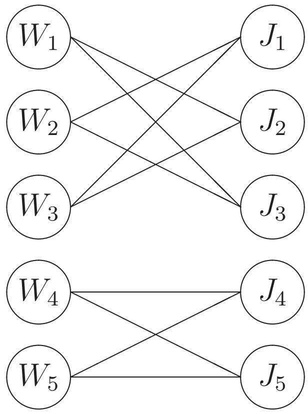


Figure 2.7: Graph-visualization of a solution (x, y) of an assignment MPM-problem LP_ε^{asp} with $n = 5$.

If we get to a worker where no uncolored edge remains, we are either done or completed only another sub-assignment. If we colored less than $2(n - |E_3|)$ edges, we simply have to continue the algorithm with another worker that still has uncolored edges.

All assignments (W_i, J_j) colored red we put in E_1 and the others colored green we put in E_2, respectively. This way we get two sub-assignments E_1, E_2 with $E_1 \cap E_2 = \emptyset$. Finally $E_1 \cup E_3$ and $E_2 \cup E_3$ represent two complete assignments of all n workers to all n jobs.

From $(|E_1| + |E_2| + 2 \cdot |E_3|) = 2n$ and (2.6) and (2.7) we directly get *(iv)*. Thus the statements *(i)* - *(v)* are proven.

Concerning *(vi)*, every feasible solution pair (Q_1, Q_2) of assignments corresponds also to a feasible solution of LP_ε^{asp}. Thus (Q_1, Q_2) cannot be better than the pair (S_1, S_2) derived above.

□

Remark 2.3.7
If the solution (x, y) of an assignment MPM-problem LP_ε^{asp} consists of several complete sub-assignments, the decomposition into two complete assignments S_1, S_2 is not unique.

Figure 2.8 shows two possible cases for the example given in Figure 2.7. Our construction ignores the cost of the single assignments. Taking the costs into account to construct the most balanced decomposition would be a much more difficult task.

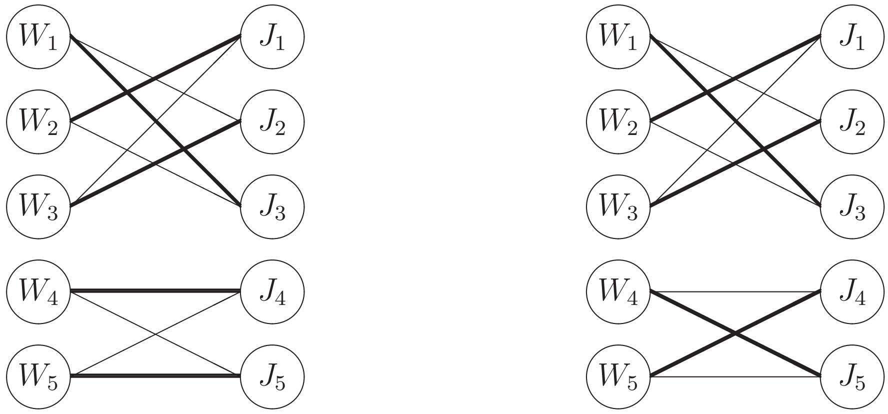


Figure 2.8: Graph-visualization of two possible decompositions of the solution (x, y) shown in Figure 2.7 into two complete assignments (S_1, S_2).

2.3.3.2 Heuristics

Here we describe briefly how to use a given local search algorithm (as described more detailed Appendix A.2) to find a locally optimal solution pair $\{\check{S}_{1(\varepsilon)}, \check{S}_{2(\varepsilon)}\}$ for the MPM-problem. The solutions can not be found one after the other. The mutual approach requires a simultaneous search for two solutions. Therefore we use the original neighborhood $\mathcal{N}$ as reference. Let $S_1, S_2 \subseteq F$ be two arbitrary solutions. Then let the new neighborhood $\mathcal{N}_{pair}$ be defined as

$$\begin{aligned} \mathcal{N}_{pair} &:= \{N_{pair}(S_1, S_2) \ : \ S_1, S_2 \in F\} \\ \text{with} \quad N_{pair}(S_1, S_2) &:= \{\{N(S_1), S_2\} \cup \{S_1, N(S_2)\}\} \end{aligned}$$

Algorithm 2.1 outlines a local search for solution pairs of an MPM-problem using the neighborhood $\mathcal{N}_{pair}$. As in the case with the penalty method (see Subsection 2.2.3) we use for the experiments the “fast” strategy with a “randomized cyclical” search order (see Appendix A.2). The initial solution pair $\{S_1^{(0)}, S_2^{(0)}\}$ is generated randomly.

Require: $P = (E, F, w)$, $\mathcal{N}_{pair}$, initial solutions $S_1^{(0)}, S_2^{(0)} \in F$

$t := 0$;
calculate $w(S_1^{(t)}, S_2^{(t)})$
while (not done) **do**
  search $\{S_1^{(t+1)}, S_2^{(t+1)}\} \in N_{pair}(S_1^{(t)}, S_2^{(t)})$ with $w(S_1^{(t+1)}, S_2^{(t+1)}) - w(S_1^{(t)}, S_2^{(t)}) < 0$
  if found **then**
    $t := t + 1$;
  else
    done;
pair $\{S_1^{(t)}, S_2^{(t)}\}$ is locally optimal for P

Algorithm 2.1: basic local search for a mutually penalized problem

Chapter 3

Experiments

3.1 Introduction

3.1.1 Idea

We consider the following situation: In a planning phase we are given a $\sum$-type problem $P = \{E, F, w\}$. We are allowed to prepare two different solutions: a first alternative S_{a_1} and a second alternative S_{a_2}. Then – in the time between the planning phase and the execution phase – the edge weights change in some random way from $\boldsymbol{w}$ to $\widehat{\boldsymbol{w}}$. Finally, in the execution phase, we choose the solution from the pair (S_{a_1}, S_{a_2}) that has the better value with respect to the new weights $\widehat{w}$.

Then we can compare the situation where we have the choice between the two alternatives S_{a_1} and S_{a_2} with the situation without choice. Let S_0 be the optimal solution of the original problem. A measure of quality is given by the performance ratio

$$\frac{\mathbb{E}\ \min\left(\widehat{w}(S_{a_1}), \widehat{w}(S_{a_2})\right)}{\mathbb{E}\ \widehat{w}(S_0)}\ .$$

The smaller this ratio, the larger is the expected use of having alternatives.

3.1.2 Random Generation of Problem Instances

In order to determine experimentally an approximation of the expected performance ratio we need a large number of different problem instances. In this subsection we describe briefly the random generation of three $\sum$-type problems we have chosen for the experiments: the shortest path problem, the assignment problem and the traveling salesman problem.

Shortest Paths in Weighted Directed Grid Graphs (SP) Consider a directed grid graph $G = (V, E)$ and a function $w : E \to \mathbb{R}^+$, assigning a length to every edge of the graph (as defined in Subsection 1.1.2.1). The lengths $w(e)$ for all $e \in E$ are independent random numbers, uniformly distributed in the interval $[0, 1] \subset \mathbb{R}$.

Assignment Problem (ASP) Consider a set of n workers $W_1, W_2, \ldots, W_n$, a set of n jobs $J_1, J_2 \ldots, J_n$ and a cost function $c : W \times J \to \mathbb{R}$ (as defined in Subsection 1.1.2.2). The costs $c_{i,j}$ for $i, j = 1, \ldots, n$ for an assignment of worker W_i to job J_j are independent random numbers, uniformly distributed in $[0, 1] \subset \mathbb{R}$.

Traveling Salesman Problem (TSP) Consider a traveling salesman problem as defined in Subsection 1.1.2.3. We generate problem instances by randomly choosing n cities, uniformly distributed in the unit square $[0, 1]^2 \subset \mathbb{R}^2$ and afterwards calculating the symmetric euclidian distances $d(c_i, c_j)$ $(i, j = 1, \ldots, n,\ i \neq j)$ between all cities.

3.1.3 Random Perturbation of the Problem Data

Definition M1 (($p; 1, c$)-model)
Consider one of the presented $\sum$-type problems $P = (E, F, w)$. We simulate the corresponding "real-world" *problem by defining perturbed weights*

$$\widehat{w}(e) := \begin{cases} \lambda_c(e) \cdot w(e) & : \text{ with probability } p \\ w(e) & : \text{ with probability } 1 - p \end{cases}$$

independently for all elements e. Here the $\lambda_c(e)$ are independent random numbers, uniformly distributed in the interval $[1, c] \subset \mathbb{R}$. We call the parameter $0 \leq p \leq 1$ **perturbation probability** *and the parameter $c \geq 1$* **perturbation width**. *This perturbation model is called $(p; 1, c)$-model.*

For fixed $p = 1$ Schwarz made experimental studies on the generation of multiple candidate solutions (see [Sch 2003]). He analyzed the situation where *all* values changed slightly (small c). We think that a generalization for arbitrary perturbation probabilities p is a promising extension of his model. So by choosing a small perturbation probability p and a large perturbation width c, we are able to simulate a situation where only a few parameters of the problem change, but these changes are typically heavy.

For the shortest path problem – as an example – the old weights w can be interpreted as ideal travel times and the new weights as the real travel times in a road network, affected by traffic densities, roadwork or weather.

3.1.4 Brief Note on the Implementation

All our experiments are implemented using C++. The algorithms we need to solve the different problems are well known. So we forego a detailed description. Partially it was possible to use freely available C++ implementations of the algorithms. Then we also mention the source.

For the experiments we need very large numbers of different instances of the problems. The random generation and the random perturbation of the problem data is done using the pseudo-random number generator from Matsumoto and Nishimura (Mersenne Twister, see [MN 1998], [MN 2002]).

Shortest Paths in Weighted Directed Grid Graphs (SP) For the shortest path problem we use only exact algorithms. For the penalty method we use an implementation of Dijkstra's algorithm (see [AMO 1993], [JM 1999]). For the mutual penalty method the linear programming approach described in Subsection 2.3.3.1 is used. The linear programs itself we solve using "`lp_solve`", an open source (mixed-integer) linear programming system that has – among others – an interface to C++ (see [BEN 2004]). For comparison we also use a k-best algorithm to compute k-shortest paths (see [Epp 1998], [JM 1999]).

Assignment Problem (ASP) We analyze the assignment problem in two different ways. First, we use exact algorithms. For the (normal) penalty method we use the hungarian method (see [MS 1991]) and for the mutual penalty method again linear programming (`lp_solve`, see [BEN 2004]). In another approach we use local search heuristics for the penalty method (see Subsections 2.2.3 and 2.3.3.2). The idea of this double analysis is to see how the penalty methods for exact algorithms and heuristics work in comparison.

Traveling Salesman Problem (TSP) The traveling salesman problem (see [Rei 1994]) is an NP-hard problem and a well known example for the successful application of local search (see [AL 2003] pp.652-673). So we again use this heuristic to compute locally optimal solutions (see Subsections 2.2.3 and 2.3.3.2).

3.2 Overview

Here we give an outline of all experiments we present in the remaining sections of this chapter.

In Section 3.3 we investigate the penalty methods with exact algorithms. First, in Subsection 3.3.1 we analyze the expected performance

$$\varphi_\varepsilon := \frac{\mathbb{E}\ \min\left(\widehat{w}(S_0), \widehat{w}(S_\varepsilon)\right)}{\mathbb{E}\ \widehat{w}(S_0)}$$

by an additional ε-penalty solution generated with the (normal) penalty method. Second, in Subsection 3.3.2 we analyze the expected performance

$$\varphi_\varepsilon^m := \frac{\mathbb{E}\ \min\left(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})\right)}{\mathbb{E}\ \widehat{w}(S_0)}$$

by the possibility to choose one of the solutions of an ε-penalty solution pair generated with the mutual penalty method. After we have compared the results of both methods in Subsection 3.3.3, we support experimentally in Subsection 3.3.4 the statement that k-best solutions are no good alternatives (see Subsection 2.1).

In Section 3.4 we investigate the penalty methods with heuristic algorithms. With heuristics we are not interested in the performance in comparison to a single locally optimal solution S_{l1}. In general, a significant performance improvement

$$\frac{\mathbb{E}\ \min\left(\widehat{w}(S_{l1}), \widehat{w}(S_{l2})\right)}{\mathbb{E}\ \widehat{w}(S_{l1})}$$

is already assured by independently generating another locally optimal solution S_{l2}. It would not be clear how big the part of the penalty methods is if we consider the ratio as

$$\frac{\mathbb{E}\ \min\left(\widehat{w}(S_{l1}), \widehat{w}(S_\varepsilon)\right)}{\mathbb{E}\ \widehat{w}(S_{l1})} \quad \text{or} \quad \frac{\mathbb{E}\ \min\left(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})\right)}{\mathbb{E}\ \widehat{w}(S_{l1})}\ .$$

Thus we directly analyze the relative improvement in comparison to a pair of independently generated locally optimal solutions S_{l1} and S_{l2}. With the penalty method (Subsection 3.4.1) we analyze

$$\chi_\varepsilon := \frac{\mathbb{E}\ \min\left(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)\right)}{\mathbb{E}\ \min\left(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2})\right)}$$

and with the mutual penalty method (Subsection 3.4.2) we analyze

$$\chi_\varepsilon^m := \frac{\mathbb{E}\ \min\left(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})\right)}{\mathbb{E}\ \min\left(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2})\right)}\ .$$

In Subsection 3.4.3 we compare the results of both heuristic methods.

For all cases we analyze the dependencies on problem parameter n and perturbation parameters p and c. For all these experiments we use the $(p; 1, c)$ perturbation model presented in Subsection 3.1.3.

In Section 3.5 we conclude this chapter with a statistical analysis of the presented results.

In Chapter 4 we report results of additional investigations (see page 57). There we do not present the results in such a scope as in this chapter.

3.3 Results of the Experiments with Exact Algorithms

3.3.1 The Penalty Method – Relative Improvement by an Additional ε-Penalty Alternative

In this subsection we focus on the case that we have an ε-penalty solution S_ε additionally to the original optimal solution S_0. We analyze the average relative improvement by alternative ε-penalty solutions with respect to the penalty parameter ε.

3.3.1.1 The Influence of the Penalty Parameter ε

For a randomly generated $\sum$-type problem $P = (E, F, w)$ we first calculate a globally optimal solution S_0. Then, for an arbitrary $\varepsilon > 0$, the penalty method gives us the corresponding alternative solution S_ε. We can take influence on this solution only by choosing the penalty parameter ε.

We calculate a whole set of ε-penalty solutions $\{S_\varepsilon\}_{\varepsilon \in I_\varepsilon}$ for different parameters $\varepsilon \in I_\varepsilon := \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$ with $0 < \varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_N$. We get N solution pairs

$$\{S_0, S_{\varepsilon_1}\}, \{S_0, S_{\varepsilon_2}\}, \ldots, \{S_0, S_{\varepsilon_N}\}$$

and can calculate the values:

$$\widehat{w}(S_0) \qquad \text{and} \qquad \min\left(\widehat{w}(S_0), \widehat{w}(S_{\varepsilon_i})\right) \text{ for } i = 1, \ldots, N\ .$$

Since we are interested in the average case, we have to calculate mean values over several runs. Each run $t = 1, 2, \ldots, T$ consists of the following four steps:

1. randomly generate a problem instance $P^{(t)} = (E, F, w^{(t)})$
2. calculate the solution pairs $\{S_0^{(t)}, S_{\varepsilon_1}^{(t)}\}, \{S_0^{(t)}, S_{\varepsilon_2}^{(t)}\}, \ldots, \{S_0^{(t)}, S_{\varepsilon_N}^{(t)}\}$
3. randomly generate an instance of perturbed weights $\widehat{w}^{(t)}$
4. evaluate the solution pairs with respect to $\widehat{w}^{(t)}$

Then the average rates $\bar{\varphi}_{\varepsilon_i}$ over the whole set of instances are

$$\bar{\varphi}_{\varepsilon_i} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_{\varepsilon_i}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_0^{(t)})} \quad \text{for } i = 1, \ldots, N \ .$$

The following figures show typical results of the computer experiments. For every tested combination of the model parameters we made $T = 10^5$ independent runs. Beside the optimal solution S_0 we computed $N = 30$ alternative solutions $S_{\varepsilon_1}, S_{\varepsilon_2}, \ldots, S_{\varepsilon_{30}}$, with $\varepsilon_1 = 0.025, \varepsilon_2 = 0.050, \ldots, \varepsilon_{30} = 0.750$.

For the shortest path problem, results for different perturbation widths c and a fixed perturbation probability p are shown. Vice versa, for the assignment problem results for a fixed width c and different probabilities p are shown. All results are for problem size $n = 25$. Figures for other problem sizes look very similar.

Figures 3.1 and 3.2 show that $\bar{\varphi}_\varepsilon$ first decreases then increases for growing ε. In our experiments ($T = 10^5$ simulation runs for each combination of $p \in \{0.01, 0.10, 0.25, 1.00\}$ and $c \in \{4, 8, 16, 32\}$) there always existed an i^* such that

$$\bar{\varphi}_{\varepsilon_1} \geq \bar{\varphi}_{\varepsilon_2} \geq \ldots \geq \bar{\varphi}_{\varepsilon_{i^*}} \leq \ldots \leq \bar{\varphi}_{\varepsilon_N} \ . \tag{3.1}$$

The appropriate globally optimal penalty parameter (which must be near ε_{i^*}, if (3.1) holds) is denoted by ε_*. For simplification, we call ε_{i^*} the *optimal penalty parameter*. Doing this, we are of course aware that neither unimodality nor the existence of an optimal ε_* are proved, but are only a conjecture from many experimental runs.

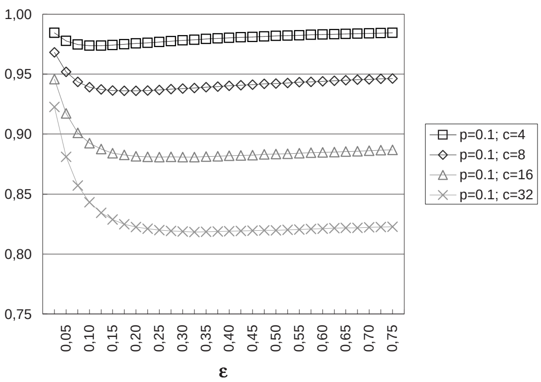


Figure 3.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ T = 10^5]$

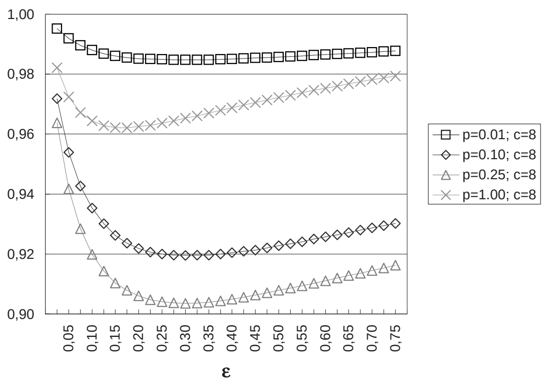


Figure 3.2: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ T = 10^5]$

Figure 3.1 also shows that larger perturbation widths c result in better rates $\bar{\varphi}_\varepsilon$. Figure 3.2 shows average rates for different perturbation probabilities p when c is fixed. There seems to be no monotonicity in p. A larger perturbation probability p does not always mean a better rate $\bar{\varphi}_\varepsilon$.

Figures 3.3 and 3.4 show the relative part of simulation runs in which the alternative ε-penalty solution is strictly better than the original optimal solution S_0:

$$r_\varepsilon^< := \frac{1}{T} \sum_{t=1\ldots T} 1_{(\widehat{w}^{(t)}(S_\varepsilon^{(t)}) < \widehat{w}^{(t)}(S_0^{(t)}))} .$$

Figures 3.5 and 3.6 show the relative part of simulation runs in which the alternative ε-penalty solution and the original optimal solution S_0 are equally good:

$$r_\varepsilon^= := \frac{1}{T} \sum_{t=1\ldots T} 1_{(\widehat{w}^{(t)}(S_\varepsilon^{(t)}) = \widehat{w}^{(t)}(S_0^{(t)}))} .$$

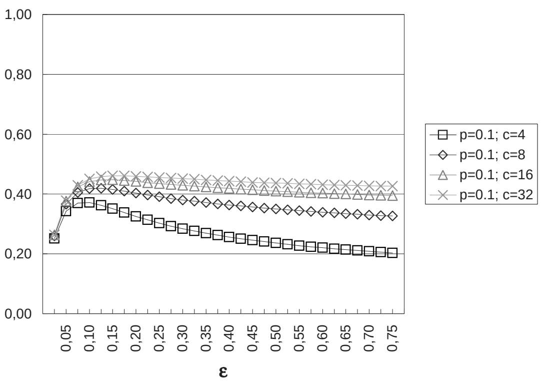


Figure 3.3: Shortest path problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$; $[n = 25;\ T = 10^5]$

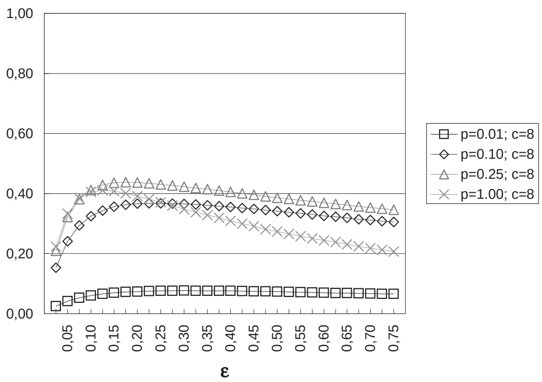


Figure 3.4: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$; $[n = 25;\ T = 10^5]$

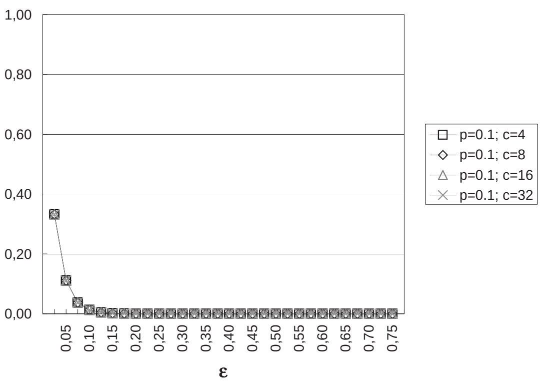


Figure 3.5: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$; $[n = 25;\ T = 10^5]$

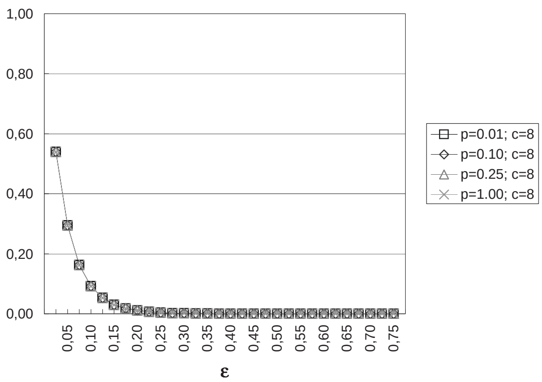


Figure 3.6: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$; $[n = 25;\ T = 10^5]$

With very small values for the penalty parameter ε, the penalty method often generates the original optimal solution S_0 itself. Then the ε-penalty alternative S_ε and the original optimal solution S_0 obviously have the same value (see Figures 3.5 and 3.6). In the remaining cases both solutions are very similar, so the probability that the ε-penalty alternative is better is still small (see Figures 3.3 and 3.4). With increasing ε the diversity of S_0 and S_ε increases. The number of runs $r_\varepsilon^<$ in which the alternative is better than S_0 increases, too. But with increasing ε the value of the alternative S_ε with respect to the original weight w gets worse. Since the edge-weight perturbation $w \to \widehat{w}$ is edge-independent, this is true on average also for $\widehat{w}$. Therefore, when ε continues to increase, the number of runs $r_\varepsilon^<$ in which the alternative is better decreases again.

For the shortest path problem with perturbation probability $p = 0.1$ and width $c = 32$ the perturbations cause the alternative S_ε to be at least equally good in more than 40% of the simulation runs. For the assignment problem with $p = 0.01$ and $c = 8$, this happens only in about 10% of the simulation runs.

For both optimization problems tested we made qualitatively the same observations:

- It is significantly advantageous to have two candidate solutions instead of having only the optimal solution of the idealized original problem.
- In our experiments there always existed an optimal intermediate penalty parameter ε_* where the benefit of using the penalty method became maximal.

The results show that the observations made by Schwarz (see [Sch 2003]) for a fixed perturbation probability $p = 1$ remain true for arbitrary perturbation probabilities $0 < p \leq 1$. Beyond this, we observed that the rates $\bar{\varphi}_\varepsilon$ are not monotone in p, instead they seem to be unimodal (see Figure 3.2). So the advantage of having two candidate solutions is highest if not too few and not too many parameters of a problem are perturbed. In the following subsection we present more detailed results on $\varepsilon_*(p, c, n)$.

3.3.1.2 The Optimal Penalty Parameter ε_*

We repeat the experiments for larger sets of simulation parameters p and c, but this time we present only the optimal penalty parameter ε_* and the corresponding average rate $\bar{\varphi}_{\varepsilon_*}$ of the optimal solution pair.

Dependence of ε_* on Perturbation Probability p and Width c

We made $T = 10^5$ runs for each combination of $c \in \{4, 8, 16, 32, 64\}$ and

$p \in \{0.01,\ 0.02,\ 0.04,\ 0.08,\ 0.16,\ 0.32,\ 0.64,\ 1.00\}$ for $n = 25$. Together with the optimal solution S_0 for each problem we computed $N = 100$ alternative solutions $S_{\varepsilon_1}, S_{\varepsilon_2}, \ldots, S_{\varepsilon_{100}}$, for $\varepsilon_1 = 0.025, \varepsilon_2 = 0.05, \ldots, \varepsilon_{100} = 2.50$.

Figures 3.7 and 3.8 show the dependence of ε_* on the perturbation probability p for different values of c.

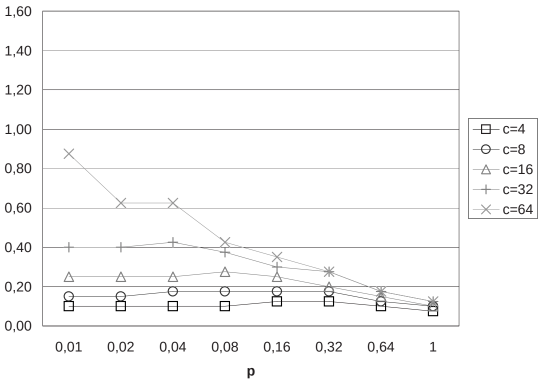


Figure 3.7: Shortest path problem – the optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};$ $T = 10^5]$

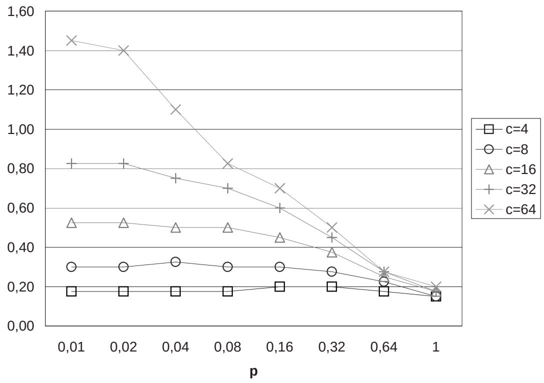


Figure 3.8: Assignment problem – the optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};$ $T = 10^5]$

Schwarz already observed for the shortest path problem that the optimal penalty parameter ε_* is weakly monotonically increasing in the perturbation width c. This remains true for all tested values of p. But the larger the fixed perturbation probability p, the smaller the growth rates of $\varepsilon_*(c)$.

At first sight the optimal penalty parameter ε_* seems to decrease monotonically for increasing perturbation probability p. Looking at the data again we observe that the optimal penalty parameter ε_* is approximately unimodal in p (see Figures 3.7 and 3.8). It first increases very slightly, then it decreases. The larger the perturbation width c, the larger are the decrease rates of $\varepsilon_*(p)$ within the decreasing segment. For small values of c (≤ 8) the optimal penalty parameter ε_* is almost constant in p.

The average rates $\bar{\varphi}_{\varepsilon_*}(p, c)$ – shown in Figures 3.9 and 3.10 – are monotonically decreasing in c and unimodal in p. The values of p where the average rates are minimal are decreasing monotonically in c.

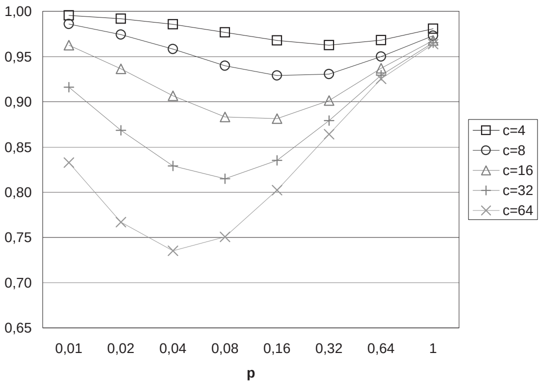


Figure 3.9: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^5]$

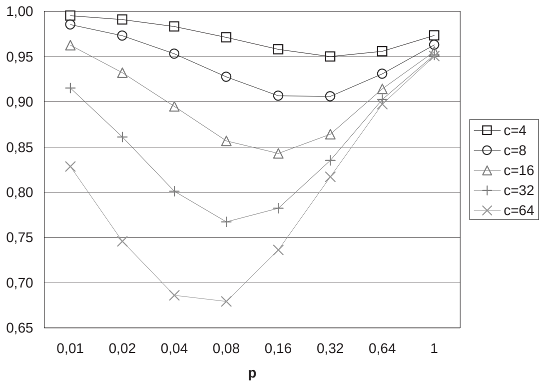


Figure 3.10: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^5]$

Dependence of ε_* on the Problem Size n

We made another 10^5 runs where we computed solution pairs for the different problem sizes $n \in \{20, 40, 60, 80, 100\}$.

		n				
		20	40	60	80	100
SP	$\varepsilon_*(n)$	0.37	0.30	0.20	0.18	0.18
	$\bar{\varphi}_{\varepsilon_*}(n)$	0.80	0.86	0.88	0.90	0.91
ASP	$\varepsilon_*(n)$	0.68	0.50	0.40	0.33	0.28
	$\bar{\varphi}_{\varepsilon_*}(n)$	0.74	0.81	0.85	0.87	0.89

Table 3.1: The optimal penalty parameters $\varepsilon_*(n)$ and the average time/cost rates $\bar{\varphi}_{\varepsilon_*}(n)$ for the shortest path problem (SP) and the assignment problem (ASP); $[p = 0.1,\ c = 32;\ I_\varepsilon = \{0.025, 0.05, \ldots, 1.5\};\ T = 10^5]$

Table 3.1 shows a decrease of the optimal penalty parameter ε_* for increasing problem size n. While this decrease happens for both the shortest path and the assignment problem, there are quantitative differences. For the assignment problem the optimal ε_* are significantly larger.

Table 3.1 also shows that the average rates $\bar{\varphi}_{\varepsilon_*}(n)$ increase for growing problem size n. For the shortest path problem we save on average about 20% for $n = 20$ and about only 9% for $n = 100$. For the assignment problem we save on average about 25% for $n = 20$ and only about 11% for $n = 100$.

3.3.2 The Mutual Penalty Method – Relative Improvement by ε-Penalty Solution Pairs

Here we analyze the case that we are allowed to choose between the two solutions from an ε-penalty solution pair $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ generated with the mutual penalty method. The original optimal solution S_0 is used only as reference.

3.3.2.1 The Influence of the Penalty Parameter ε

We repeat the experiments from Subsection 3.3.1 with the mutual penalty method. Again we calculate mean values over several runs. The average rates $\bar{\varphi}^m_{\varepsilon_i}$ over the whole set of instances are

$$\bar{\varphi}^m_{\varepsilon_i} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S^{(t)}_{1(\varepsilon_i)}), \widehat{w}^{(t)}(S^{(t)}_{2(\varepsilon_i)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S^{(t)}_0)} \quad \text{for } i = 1, \ldots, N \ .$$

The following figures show typical results of the computer experiments. We computed $N = 30$ solution pairs $\{S_{1(\varepsilon_1)}, S_{2(\varepsilon_1)}\}, \{S_{1(\varepsilon_2)}, S_{2(\varepsilon_2)}\}, \ldots, \{S_{1(\varepsilon_{30})}, S_{2(\varepsilon_{30})}\}$, for $\varepsilon_1 = 0.025, \varepsilon_2 = 0.050, \ldots, \varepsilon_{30} = 0.750$. Since the computation time of the mutual penalty method is significantly larger in comparison to the penalty method we could do (only) $T = 10^4$ independent runs for every tested combination of the model parameters.

For the shortest path problem, results for different perturbation widths c and a fixed perturbation probability p are shown. Vice versa, for the assignment problem results for a fixed width c and different probabilities p are shown. All results are for problem size $n = 25$. Figures for other problem sizes look very similar.

Figures 3.11 and 3.12 show that $\bar{\varphi}^m_{\varepsilon}$ first decreases then increases for growing ε as for the case with the penalty method. In all our experiments ($T = 10^4$ simulation runs for each combination of $p \in \{0.01, 0.10, 0.25, 1.00\}$ and $c \in \{4, 8, 16, 32\}$) there existed an optimal penalty parameter ε_{i^*} such that

$$\bar{\varphi}^m_{\varepsilon_1} \geq \bar{\varphi}^m_{\varepsilon_2} \geq \ldots \geq \bar{\varphi}^m_{\varepsilon_{i^*}} \leq \ldots \leq \bar{\varphi}^m_{\varepsilon_N}.$$

With the mutual penalty method the average rate $\bar{\varphi}^m_{\varepsilon}$ can be larger than 1. So, if the penalty parameter ε is badly chosen the possibility to choose from two candidate solutions is worthless on average. It is even worse than the reference case where we have only the original optimal solution S_0.

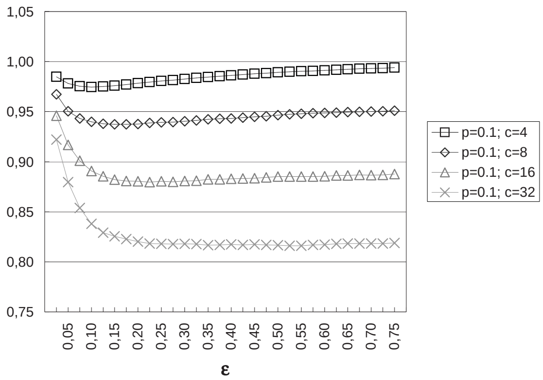


Figure 3.11: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ T = 10^4]$

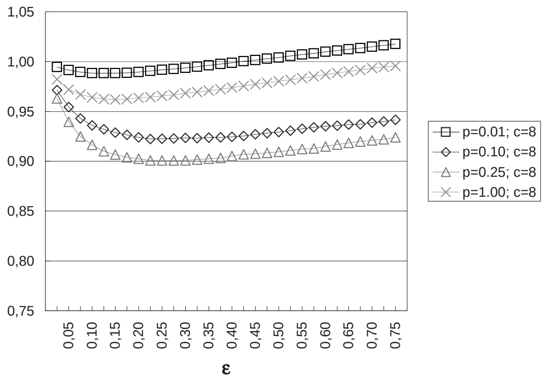


Figure 3.12: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ T = 10^4]$

Figures 3.13 and 3.14 show the relative part of simulation runs $r_{\varepsilon}^{<}$ where (at least) one of the solutions of the ε-penalty pair $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ is better than the original optimal solution S_0:

$$r_{\varepsilon}^{<} := \frac{1}{T} \sum_{t=1 \ldots T} 1_{(\min(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}), \widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)})) < \widehat{w}^{(t)}(S_0^{(t)}))}$$

In all cases the frequency $r_{\varepsilon}^{<}$ first increases and then decreases with increasing penalty parameter ε. For many penalty parameters ε the frequency of an improvement $r_{\varepsilon}^{<}$ is significantly smaller than 1/2. Nevertheless the according average rates $\bar{\varphi}_{\varepsilon}^{m}$ are smaller than 1.

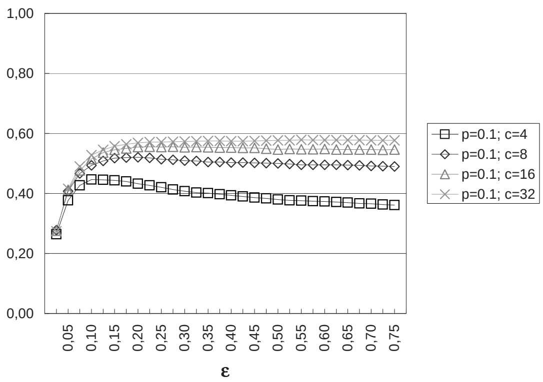


Figure 3.13: Shortest path problem – relative part of simulation runs $r_{\varepsilon}^{<}$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ T = 10^4]$

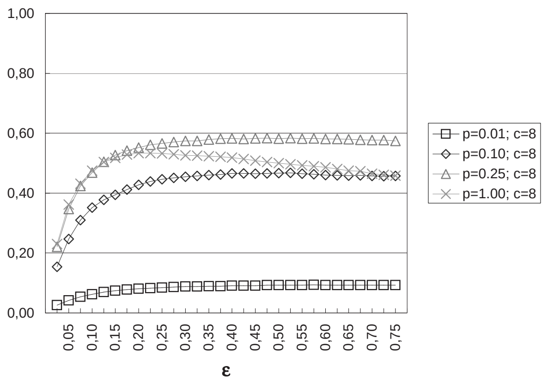


Figure 3.14: Assignment problem – relative part of simulation runs $r_{\varepsilon}^{<}$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ T = 10^4]$

This results from a large relative part of simulation runs

$$r_\varepsilon^= := \frac{1}{T}\sum_{t=1\ldots T} 1_{(\min(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}),\widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)}))=\widehat{w}^{(t)}(S_0^{(t)}))}$$

where the better solution from the pair $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ has the same value as the original optimal solution S_0 (see Figures 3.15 and 3.16). Further analysis showed that in nearly all of these cases the solutions have not only the same value but are indeed identical.

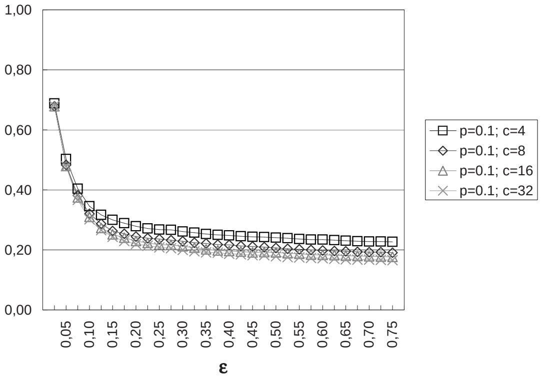


Figure 3.15: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ T = 10^4]$

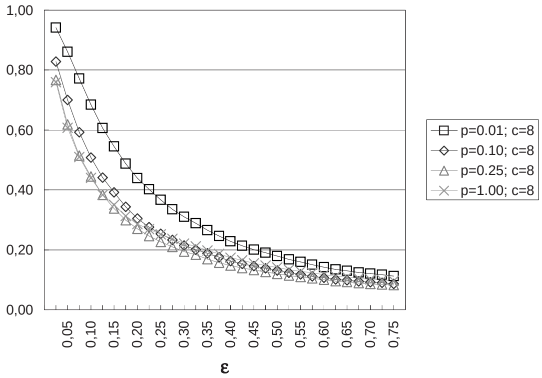


Figure 3.16: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ T = 10^4]$

3.3.2.2 The Optimal Penalty Parameter ε_*^m

Again we have a closer look at the optimal penalty parameter ε_*^m and the corresponding average rate $\bar{\varphi}_{\varepsilon_*}^m$ of the optimal solution pair.

Dependence of ε_*^m on the Perturbation Probability p and the Width c

We made $T = 10^4$ runs for each combination of $p \in \{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.00\}$ and $c \in \{4, 8, 16, 32, 64\}$ for $n = 25$. Together with the optimal solution S_0 for each problem we computed $N = 100$ solution pairs $\{S_{1(\varepsilon_1)}, S_{2(\varepsilon_1)}\}$, $\{S_{1(\varepsilon_2)}, S_{2(\varepsilon_2)}\}, \ldots, \{S_{1(\varepsilon_{100})}, S_{2(\varepsilon_{100})}\}$ for $\varepsilon_1 = 0.025, \varepsilon_2 = 0.05, \ldots, \varepsilon_{100} = 2.50$.

Figures 3.17 and 3.18 show the dependence of ε_*^m on the perturbation probability p for different values of c.

We were not able to do as much simulation runs T as with the penalty method. So the experimentally observed average optimal penalty parameters ε_*^m do not look as smooth. Furthermore, the number of runs T necessary to get smooth looking

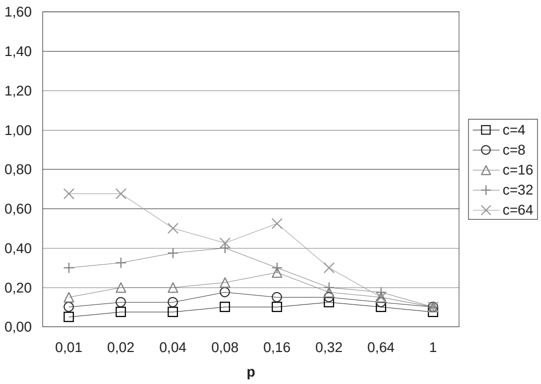


Figure 3.17: Shortest path problem – the optimal penalty parameters $\varepsilon_*^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^4]$

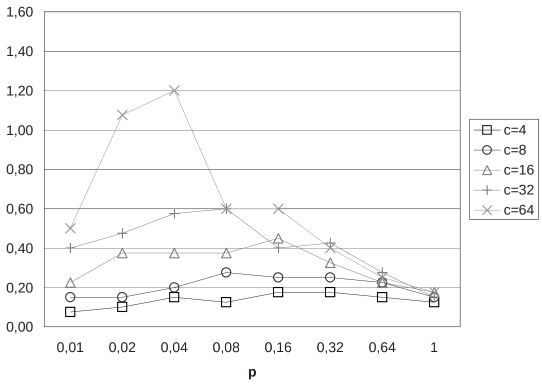


Figure 3.18: Assignment problem – the optimal penalty parameters $\varepsilon_*^m(p,c)$. $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^4]$

results depends on the simulation parameters p and c. Obviously the larger the perturbation width c, the larger the variation of the results. If we also decrease the perturbation probability p, the variation increases again. So the results shown in Figures 3.17 and 3.18 for very small values of p and large c are not as smooth as for larger p and smaller c.

Anyhow we can see that the dependencies are qualitatively quite similar to the case with the penalty method (see Figures 3.7 and 3.8). The optimal penalty parameter ε_*^m seems to be weakly monotonically increasing in the perturbation width c and unimodal in p. It first increases slightly then it decreases with increasing p.

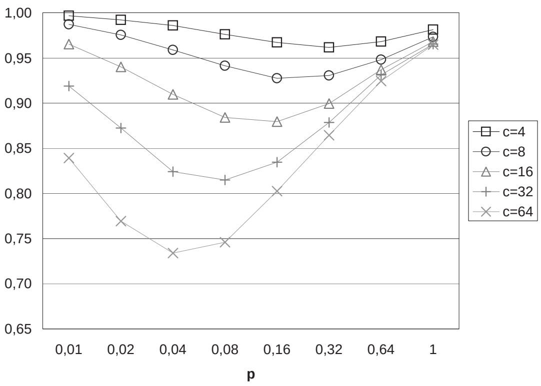


Figure 3.19: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon_*}^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^4]$

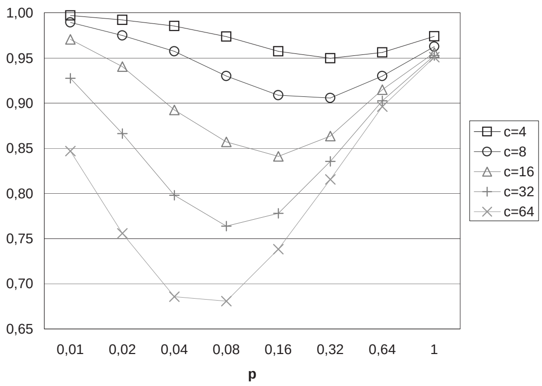


Figure 3.20: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon_*}^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^4]$

The observed average rates $\bar{\varphi}_{\varepsilon_*}^m(p,c)$ for different parameters p and c (see Figures 3.19 and 3.20) are nearly identical to those observed with the penalty method (see Figures 3.9 and 3.10). There are no significant differences.

Dependence of ε_*^m on the Problem Size n

We made another 10^4 runs where we computed solution pairs for the different problem sizes $n \in \{10, 15, 20, 25, 30\}$.

Table 3.2 shows that the optimal penalty parameter ε_*^m decreases for increasing problem size n. It also shows that the average rates $\bar{\varphi}_{\varepsilon_*}^m(n)$ increase for growing problem size n. We were not able to analyze the same set of problem sizes n as for the penalty method. But probably we would not get results that are very different in comparison to those observed with the penalty method.

		n				
		10	15	20	25	30
SP	$\varepsilon_*^m(n)$	0.68	0.53	0.38	0.28	0.35
	$\bar{\varphi}_{\varepsilon_*}^m(n)$	0.73	0.77	0.80	0.82	0.83
ASP	$\varepsilon_*^m(n)$	0.90	0.88	0.55	0.55	0.53
	$\bar{\varphi}_{\varepsilon_*}^m(n)$	0.66	0.71	0.74	0.76	0.78

Table 3.2: The optimal penalty parameters $\varepsilon_*^m(n)$ and the average time/cost rates $\bar{\varphi}_{\varepsilon_*}^m(n)$ for the shortest path problem (SP) and the assignment problem (ASP);
[$p = 0.1$, $c = 32$; $I_\varepsilon = \{0.025, 0.05, \ldots, 1.5\}$; $T = 10^4$]

3.3.3 Concluding Comparison

In all our experiments it turned out that it is clearly advantageous to have two different solutions to choose from. In the case of seldom but severe perturbations the benefits are most impressive.

But these benefits get only realized if a good penalty parameter ε is chosen. In our experiments there always existed an optimal intermediate penalty parameter $\varepsilon_*^{(m)}$ for which the benefit was maximal. The more the penalty parameter in use differed from the optimal penalty parameter the worse the average results were. Unfortunately (in view of practice), the optimal penalty parameter $\varepsilon_*^{(m)}$ depends on the problem, the problem size n and the simulation parameters p and c. The following rules of thumb give a general idea of the dependencies: The optimal penalty parameter $\varepsilon_*^{(m)}$

- monotonically increases in the perturbation width c.
- first increases slightly then decreases in the perturbation probability p.
  For small perturbation widths c the penalty parameter ε_* is nearly constant in p.
- monotonically decreases in the problem size n.

Whereas the qualitative dependencies of the optimal penalty parameter are equal for the normal and the mutual penalty method, they differ a bit quantitatively. For the mutual approach the optimal penalty parameter ε_*^m tends to be a bit smaller than ε_*.

In the experiments so far the **mutual** penalty method gains no extra benefit in comparison to the **normal** penalty method. At first sight both approaches are quite different. But the experimental results are very similar. Figures 3.21 and 3.22 show that the generated solution pairs of both methods ($\{S_0, S_\varepsilon\}$ and $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$) are not as different as supposed, at least not for our class of randomly generated problems with uniformly distributed data. The weighted common part

$$v_{(S_{1(\varepsilon)},S_{2(\varepsilon)})} := \frac{1}{T}\sum_{t=1}^{T} \frac{\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)} \cap S_{2(\varepsilon)}^{(t)})}{\widehat{w}^{(t)}(S_0^{(t)})}$$

of the solutions of the ε-penalty pair gets really small for larger penalty parameters ε. But the maximal common part

$$v_{(S_0,\{S_{1(\varepsilon)},S_{2(\varepsilon)}\})}^{max} := \frac{1}{T}\sum_{t=1}^{T} \frac{\max(\widehat{w}^{(t)}(S_0^{(t)} \cap S_{1(\varepsilon)}^{(t)}), \widehat{w}^{(t)}(S_0^{(t)} \cap S_{2(\varepsilon)}^{(t)}))}{\widehat{w}^{(t)}(S_0^{(t)})}$$

of one of the solutions $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ with the original optimal solution S_0 stays relatively high at about 80%. This means that very often one of the solutions from the ε-penalty pair is similar to the original optimal solution.

$\triangle - v_{(S_{1(\varepsilon)},S_{2(\varepsilon)})}$

$\square - v_{(S_0,\{S_{1(\varepsilon)},S_{2(\varepsilon)}\})}^{max}$

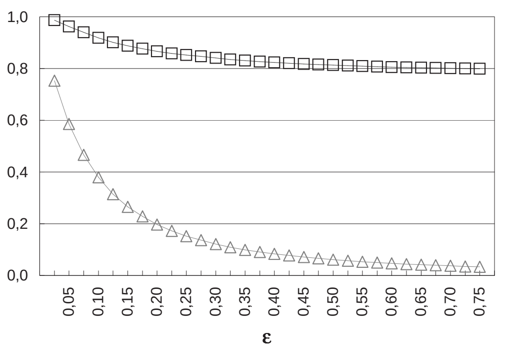


Figure 3.21: Shortest path problem – common parts of the solutions; $[n = 25;\ T = 10^4]$

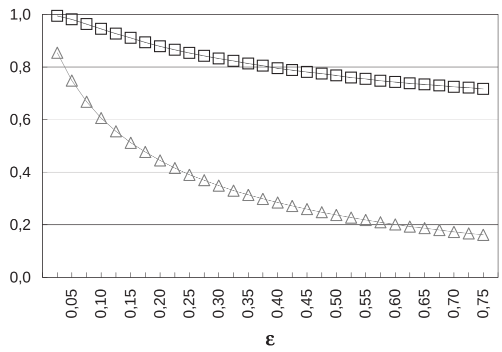


Figure 3.22: Assignment problem – common parts of the solutions; $[n = 25;\ T = 10^4]$

Furthermore, it is not clear whether the observations we made so far depend on the $(p; 1, c)$-perturbation model we used. Another perturbation model – for example with perturbations that are not independent of the edges – could change the observations. At this point we skip this question and postpone it to a later section where we test some other perturbation models (see 4.2).

3.3.4 Penalty Method vs. k-th Best Approach for Shortest Paths

Using a 2-best algorithm is probably the most intuitive way to get a pair of solutions. One simply calculates the best and the 2nd-best solution. We repeat the experiment for the shortest path problem using this approach. Together with the optimal solution S_0 we compute the next best solution S_1. Then we calculate the $\widehat{w}$-length of the optimal solution S_0 and the minimum of the $\widehat{w}$-lengths of the pair $\{S_0, S_1\}$. We make this for T rounds and calculate the average rate $\bar{\varphi}_2$.

$$\bar{\varphi}_2 := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_1^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_0^{(t)})}$$

Figure 3.23 shows the average time rates $\bar{\varphi}_2(p, c)$ of the 2-best approach. Figure 3.24 shows again the average time rates $\bar{\varphi}_{\varepsilon_*}(p, c)$ using the penalty method with the optimal penalty parameter $\varepsilon_*(p, c)$.

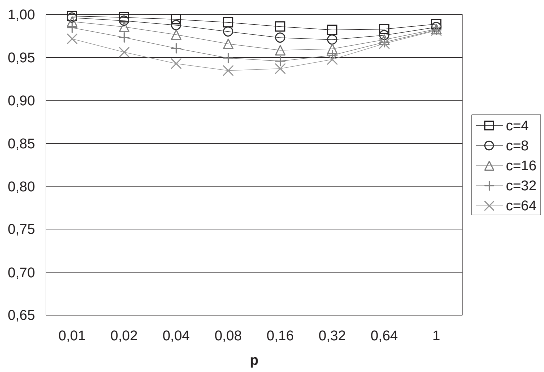


Figure 3.23: 2-best approach – average time rates $\bar{\varphi}_2(p, c)$; $[n = 25;\ T = 10^5]$

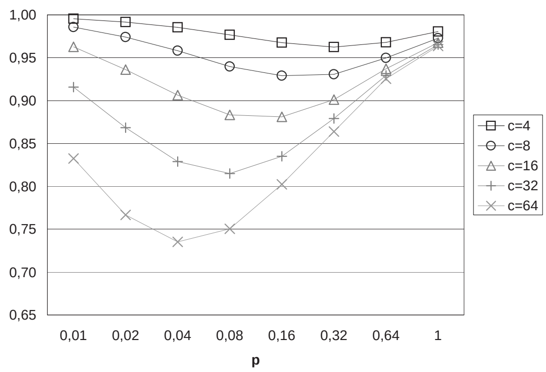


Figure 3.24: Penalty method – average time rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ T = 10^5]$

In general, the best and the 2nd best solution will be very similar to each other, differing only in very few details. So the average rates we see in Figure 3.23 are quite bad in comparison to the ones of the penalty method shown in Figure 3.24.

We generalize this approach by using the k-th best solution S_{k-1} instead of the 2nd best solution S_1 as the second part of a solution pair. We choose a k and get the pair $\{S_0, S_{k-1}\}$. We make another experiment to see which k to choose. Together with the optimal solution S_0 we compute all next best solutions $S_1, S_2, \ldots, S_{N-1}$. Then we calculate the minima of the $\widehat{w}$-lengths of all the pairs $\{S_0, S_1\}, \{S_0, S_2\}, \ldots, \{S_0, S_{N-1}\}$. We make this for T runs and calculate the average rates $\bar{\varphi}_k$ for $k = 2, 3, \ldots, N$.

$$\bar{\varphi}_k := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_{k-1}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_0^{(t)})}$$

Figure 3.25 shows the results of $T = 10^5$ runs for problem size $n = 25$. The rates $\bar{\varphi}_k$ seem to be nearly constant in k up to very large values of k.

We repeated the experiment for a smaller problem size $n = 8$. Starting with the 2nd best solution, the average rates $\bar{\varphi}_k$ of the solution pairs of the k-th best approach seem to get smaller for growing k up to some k_*. From there onwards $\bar{\varphi}_k$ increases again (see Figure 3.26). It is also possible that k_* is equal to 2, then the decreasing part is empty.

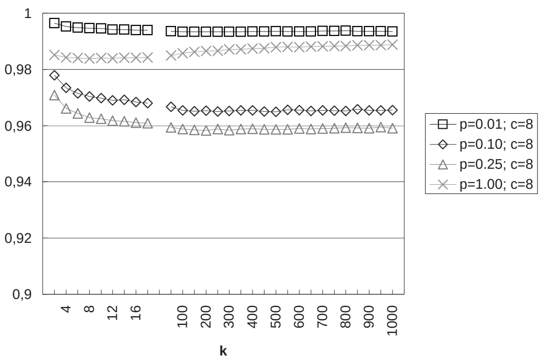


Figure 3.25: k-th best approach – average time rates $\bar{\varphi}_k$; [$n = 25$; $T = 10^5$]

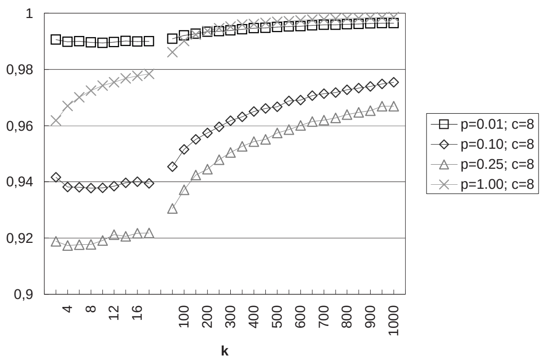


Figure 3.26: k-th best approach – average time rates $\bar{\varphi}_k$; [$n = 8$; $T = 10^5$]

We are interested in the index k_* with the smallest average rate $\bar{\varphi}_{k_*} \leq \bar{\varphi}_k$ for all $k \neq k_*$. This index seems to be very large for some combinations of the simulation parameters and the problem size. As an example we tested problem size $n = 10$ with perturbation probability $p = 0.01$ and perturbation width $c = 256$. The optimal index k_* for this setting was about 32000. So we have chosen a rather small problem size $n = 8$ to be able to calculate all possible paths up to the maximal $k = 3431$ within acceptable time. We made $T = 10^5$ runs for each combination of the real-world simulation parameters $p \in \{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.00\}$ and $c \in \{4, 8, 16, 32, 64\}$.

Figure 3.27 shows the optimal index $k_*(p,c)$ of the k-th best approach and Figure 3.28 shows the optimal penalty parameter $\varepsilon_*(p,c)$ of the penalty method. The dependencies of $k_*(p,c)$ on the perturbation probability p and the perturbation width c are qualitatively identical to those of the optimal penalty parameter $\varepsilon_*(p,c)$. The optimal index k_* gets large for small values of p and large values of c.

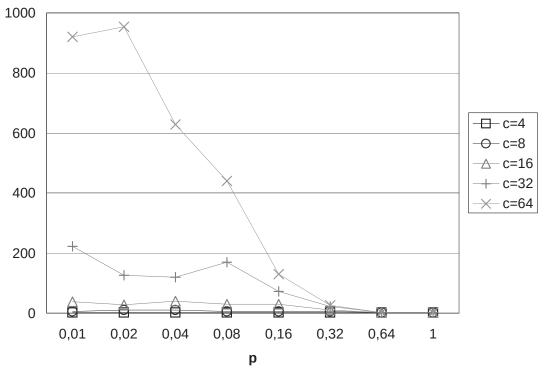


Figure 3.27: k-th best approach – the optimal indices $k_*(p,c)$; $[n = 8;\ I_k = \{1, 3, 5, \ldots, 3431\};\ T = 10^5]$

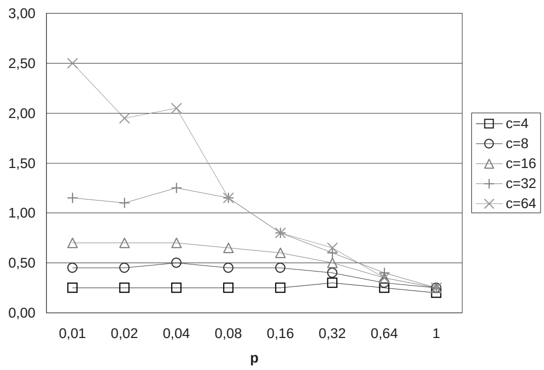


Figure 3.28: Penalty method – the optimal penalty parameters $\varepsilon_*(p,c)$; $[n = 8;\ I_\varepsilon = \{0.05, 0.10, \ldots, 10\};\ T = 10^5]$

Figures 3.29 and 3.30 show the average rates $\bar{\varphi}$ of the solution pairs. We see that those of the penalty method are significantly better than those of the k-th best approach. This is true for all tested parameter settings.

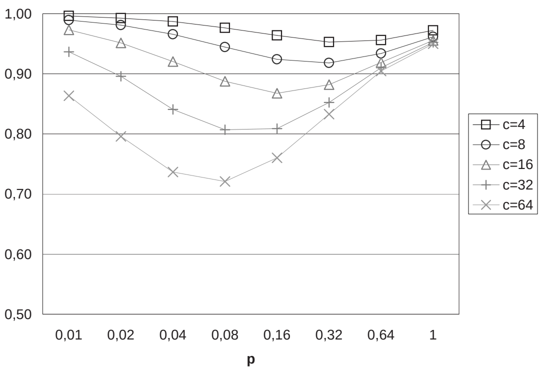


Figure 3.29: k-th best approach – average time rates $\bar{\varphi}_{k_*}(p,c)$; $[n = 8;\ I_k = \{1, 3, 5, \ldots, 3431\};\ T = 10^5]$

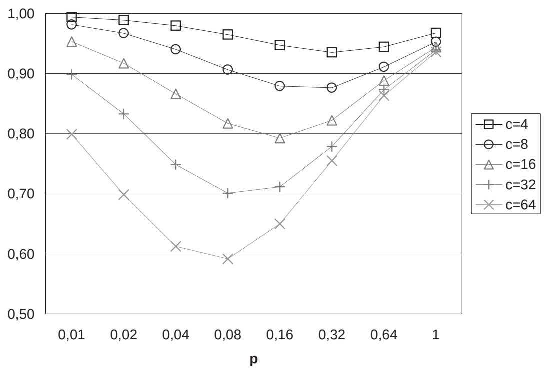


Figure 3.30: Penalty method – average time rates $\bar{\varphi}_{\varepsilon_*}(p,c)$; $[n = 8;\ I_\varepsilon = \{0.05, 0.10, \ldots, 10\};\ T = 10^5]$

We made another series of runs where we computed solution pairs for different problem sizes $n \in \{10, 15, 20, 25, 30\}$. Whereas the optimal penalty parameter decreases for growing problem size n, the optimal k_* increases rapidly (see Table 3.3). The penalty parameter does not affect the running time of an algorithm that uses the penalty method to generate an alternative. But obviously a k-best algorithm needs more time to calculate the k-th best solution for larger k.

Table 3.3 also shows the average rates $\bar{\varphi}$ of the solution pairs for growing problem size n. Again those of the penalty method are significantly better than those of the k-th best approach.

		n				
		10	15	20	25	30
k-th best approach	$k_*(n)$	490	3780	16920	29820	38820
	$\bar{\varphi}_{k_*}(n)$	0.82	0.85	0.87	0.89	0.90
penalty method	$\varepsilon_*(n)$	0.65	0.55	0.40	0.35	0.30
	$\bar{\varphi}_{\varepsilon_*}(n)$	0.72	0.77	0.80	0.82	0.83

Table 3.3: The optimal indices $k_*(n)$ and the average time rates $\bar{\varphi}_{k_*}(n)$; and the optimal penalty parameters $\varepsilon_*(n)$ and the average time rates $\bar{\varphi}_{\varepsilon_*}(n)$ for the shortest path problem; $[p = 0.1,\ c = 32;\ I_k = \{9, 19, \ldots, 39999\};\ I_\varepsilon = \{0.05, 0.10, \ldots, 5.00\};\ T = 10^5]$

In Figures 3.25 and 3.26 we have seen that the average rates $\bar{\varphi}_k$ of the k-th best approach solution pairs $\{S_0, S_k\}$ decrease very slow with growing k, even more so for larger problem sizes. Maybe the benefit of the solution pair $\{S_0, S_{k_*}\}$ with a very large k_* is not really much bigger than the benefit of a pair with a smaller value of k. So we made another $T = 10^5$ runs for problem size $n = 25$. This time we generated only the 50-th best path and calculated the average rate $\bar{\varphi}_{50}$ of the solution pair $\{S_0, S_{49}\}$. Figure 3.31 shows the results of this experiment. The average rates $\bar{\varphi}_{50}$ are significantly worse than the average rates $\bar{\varphi}_{\varepsilon_*}$ of the penalty method (see Figure 3.24). Raising the fixed index k of the second solution of the pair to $k = 500$ does not improve the results notably (see Figure 3.32).

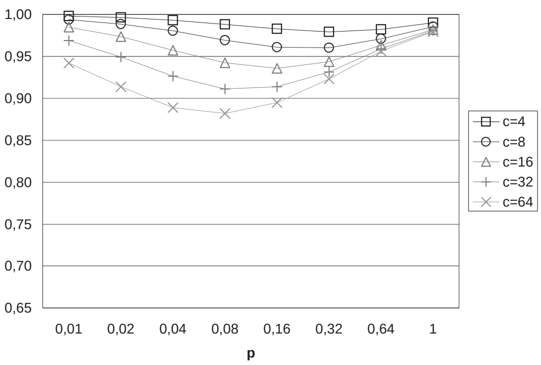


Figure 3.31: 50-th best approach – average time rates $\bar{\varphi}_{50}(p, c)$; $[n = 25;\ T = 10^5]$

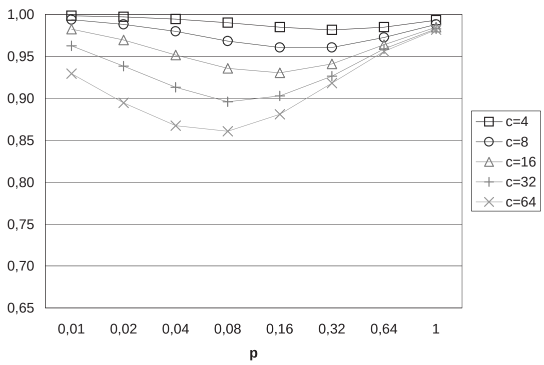


Figure 3.32: 500-th best approach – average time rates $\bar{\varphi}_{500}(p, c)$; $[n = 25;\ T = 10^5]$

The fact that the index k for generating a true alternative with a k-best algorithm is extremely large for a normal problem size makes the k-th-best approach inefficient in comparison to the penalty method. As we saw in Figure 3.29, even using the k-th best approach with the average optimal index k_* is qualitatively worse than using the penalty method.

3.4 Results of the Experiments with Heuristics

As in Section 3.3 we start with an optimization problem $P = (E, F, w)$ and are allowed to provide two different solutions. Then we learn about the simulated real-world data $\widehat{w}$ and can choose the solution that has the better value with respect to $\widehat{w}$ from our pair. Assume we are given a problem which we are not able to solve exactly within a given time, but we are able to solve the problem heuristically. We check whether it is advantageous within the model of perturbed data to use penalty methods to prepare alternative solutions with a heuristic.

Another intuitive way to generate a pair of solutions is to run a randomized heuristic twice. In general, this leads to two different, heuristic solutions.

First we want to have an idea of the ratio of the minimal value of two independent, locally optimal solutions to the value of one locally optimal solution. Let S_{l_1} and S_{l_2} be two local optima. Then we are interested in the quotient:

$$\bar{Q}(S_{l_1}, S_{l_2}) := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_{l_1}^{(t)}), \widehat{w}^{(t)}(S_{l_2}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_{l_1}^{(t)})}.$$

Tables 3.4 and 3.5 show $\bar{Q}$ for $T = 10^6$ runs for the traveling salesman problem and the assignment problem. In both cases we used problem size $n = 25$ and different combinations of the parameters p and c. (The quotas $\bar{Q}$ strongly depend on the heuristics used (see Appendix A.2).)

$\bar{Q}$	c=2	c=8	c=32	c=128
p=0.01	0.96	0.96	0.94	0.92
p=0.10	0.96	0.92	0.86	0.81
p=0.25	0.96	0.91	0.86	0.84
p=1.00	0.96	0.94	0.94	0.94

Table 3.4: Traveling salesman problem – $[n = 25;\ T = 10^6]$

$\bar{Q}$	c=2	c=8	c=32	c=128
p=0.01	0.94	0.93	0.90	0.87
p=0.10	0.94	0.90	0.82	0.75
p=0.25	0.93	0.88	0.83	0.80
p=1.00	0.93	0.92	0.91	0.91

Table 3.5: Assignment problem – $[n = 25;\ T = 10^6]$

Obviously the minimal value of two independent locally optimal solutions is on average significantly smaller than the value of only one locally optimal solution. So we examine whether a pair of solutions generated with the penalty method is on average even better than a pair of two independently generated solutions.

3.4.1 The Penalty Method – Relative Improvement by an Additional Heuristic ε-Penalty Alternative

Here we focus on the case that we have a heuristic ε-penalty solution $\check{S}_\varepsilon$ additionally to a locally optimal solution $\check{S}_{l1}$. We analyze the average relative improvement by alternative heuristic ε-penalty solutions with respect to the penalty parameter ε.

3.4.1.1 The Influence of the Penalty Parameter ε

In the following experiments we calculate beside a locally optimal solution $\check{S}_{l_1}$ a set of heuristic ε-penalty solutions $\check{S}_{\varepsilon_1}, \check{S}_{\varepsilon_2}, \ldots, \check{S}_{\varepsilon_N}$ for different parameters $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N$ and another independently generated locally optimal solution $\check{S}_{l_2}$. We get N solution pairs $\{\check{S}_{l_1}, \check{S}_{\varepsilon_1}\}, \{\check{S}_{l_1}, \check{S}_{\varepsilon_2}\}, \ldots, \{\check{S}_{l_1}, \check{S}_{\varepsilon_N}\}$ and calculate the minima of the function values of the computed pairs of solutions with respect to the simulated real-world problem:

$$\min\left(\widehat{w}(\check{S}_{l_1}), \widehat{w}(\check{S}_{\varepsilon_i})\right) \text{ for } i = 1, \ldots, N \qquad \text{and} \qquad \min\left(\widehat{w}(\check{S}_{l_1}), \widehat{w}(\check{S}_{l_2})\right)$$

Again we calculate mean values over different problem instances and different instances of perturbed weights. Let T be the number of tests for a given parameter set. The average performance rates $\bar{\chi}_{\varepsilon_i}$ are:

$$\bar{\chi}_{\varepsilon_i} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(\check{S}_{l_1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{\varepsilon_i}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(\check{S}_{l_1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{l_2}^{(t)})\right)} \quad \text{for } i = 1, \ldots, N\,.$$

The following figures show some of the results of the computer experiments. More figures are shown in Appendix B.2.1. We made $T = 10^6$ runs for each combination of the simulation parameters $p \in \{0.01, 0.1, 0.25, 1.00\}$, $c = 8$ and $p = 0.1$, $c \in \{4, 8, 16, 32\}$. We computed $N = 30$ alternative solutions $\check{S}_{\varepsilon_1}, \check{S}_{\varepsilon_2}, \ldots, \check{S}_{\varepsilon_{30}}$ with $\varepsilon_1 = 0.025, \varepsilon_2 = 0.050, \ldots, \varepsilon_{30} = 0.750$. Again we always set $n = 25$.

As in the case with exact algorithms, Figures 3.33 and 3.34 show that the average rates $\bar{\chi}_\varepsilon$ first decrease and then increase for growing ε such that

$$\bar{\chi}_{\varepsilon_1} \geq \bar{\chi}_{\varepsilon_2} \geq \ldots \geq \bar{\chi}_{\varepsilon_{i^*}} \leq \ldots \leq \bar{\chi}_{\varepsilon_N}\,.$$

So this time there always existed an optimal penalty parameter $\varepsilon_* := \varepsilon_{i_*}$, too. Since we compare the minimal value of the solution pairs $\{\check{S}_{l_1}, \check{S}_{\varepsilon_i}\}$ to the minimal

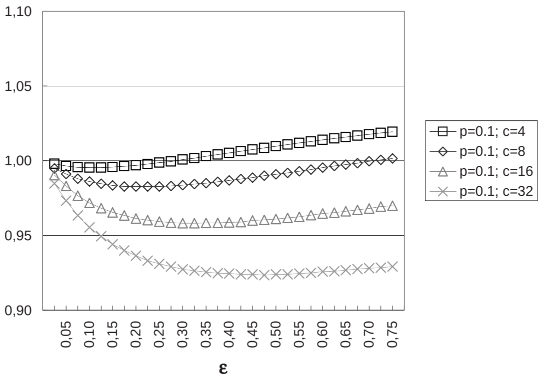


Figure 3.33: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ T = 10^6]$

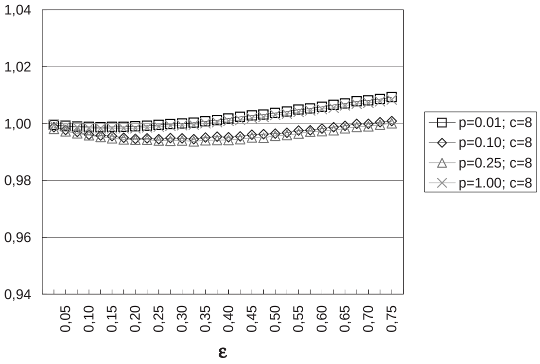


Figure 3.34: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ T = 10^6]$

value of two independently generated locally optimal solutions, the ratios $\bar{\chi}_\varepsilon$ can now exceed 1. That means that generating a solution pair with the penalty method (for a badly chosen penalty parameter ε) may be worse on average than generating two independent, locally optimal solutions. But for all settings we tested, the rate $\bar{\chi}_{\varepsilon_*}$ for the optimal penalty parameter ε_* was better (i.e. smaller) than 1.

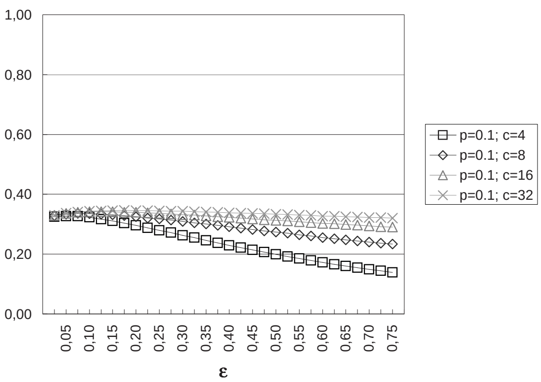


Figure 3.35: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^<$ in which the heuristic ε-penalty solution $\check{S}_\varepsilon$ is best; $[n = 25;\ T = 10^6]$

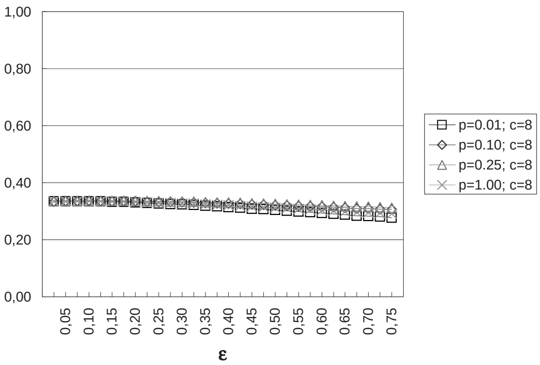


Figure 3.36: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ in which the heuristic ε-penalty solution $\check{S}_\varepsilon$ is best; $[n = 25;\ T = 10^6]$

The larger the penalty parameter ε, the worse (on average) the value of the heuristic ε-penalty solution $\check{S}_\varepsilon$ and the smaller the relative part of simulation runs

$$r_\varepsilon^< := \frac{1}{T} \sum_{t=1\ldots T} 1_{(\min(\widehat{w}^{(t)}(\check{S}_{l1}^{(t)}), \widehat{w}^{(t)}(\check{S}_\varepsilon^{(t)})) < \min(\widehat{w}^{(t)}(\check{S}_{l1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{l2}^{(t)})))}$$

$$= \frac{1}{T} \sum_{t=1\ldots T} 1_{((\widehat{w}^{(t)}(\check{S}_\varepsilon^{(t)}) < \widehat{w}^{(t)}(\check{S}_{l1}^{(t)}))\ \wedge\ (\widehat{w}^{(t)}(\check{S}_\varepsilon^{(t)}) < \widehat{w}^{(t)}(\check{S}_{l2}^{(t)})))}$$

in which the heuristic ε-penalty solution $\check{S}_\varepsilon$ is best (see Figures 3.35 and 3.36).

3.4.1.2 The Optimal Penalty Parameter ε_*

Here we present the results of the experiments for the dependence on the problem size n and the simulation parameters p and c. This time we present only the optimal penalty parameter ε_* and the according rates $\bar{\chi}_{\varepsilon_*}$. For the experiments we used the same parameter sets as for the case of exact algorithms ($p \in \{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.00\}$ and $c \in \{4, 8, 16, 32, 64\}$ for $n = 25$) but made 10^6 runs each (instead of 10^5).

Dependence of ε_* on the Perturbation Width c and the Perturbation Probability p

Exactly as for the case of exact algorithms, the optimal penalty parameter ε_* is monotonically increasing in the perturbation width c (see Figures 3.37 and 3.38). The larger the fixed perturbation probability p, the smaller the growth rates of $\varepsilon_*(c)$. For $p = 1$ the penalty parameter ε_* is nearly constant in c. For the dependence on the perturbation probability p we observe that the optimal penalty parameter ε_* is approximately unimodal in p as for the exact case. It first increases slightly, then it decreases. The larger the perturbation width c, the larger the decrease rates of $\varepsilon_*(p)$ (within the decreasing segment). For small values of c (≤ 8) the optimal penalty parameter ε_* is almost constant in p.

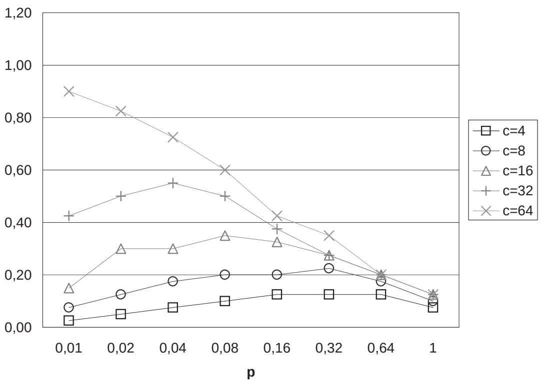


Figure 3.37: Traveling salesman problem – the optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^6]$

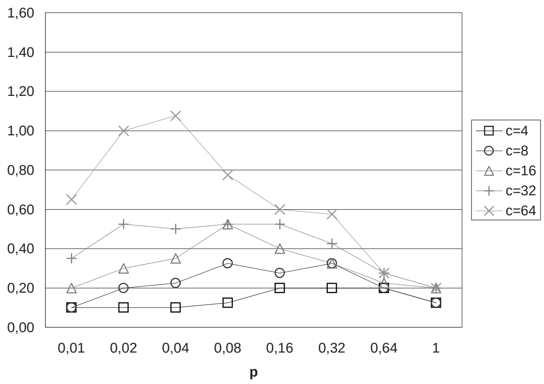


Figure 3.38: Assignment problem – the optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^6]$

The qualitative behavior of the rates $\bar{\chi}_{\varepsilon^*}$ is also similar to the exact case (see Figures 3.39 and 3.40). The values are monotonically decreasing in c and unimodal in p. The values of p where the rates $\bar{\chi}_{\varepsilon_*}$ are minimal are monotonically increasing in c. The only significant difference between the exact and the heuristic case is the size of the rates $\bar{\varphi}_{\varepsilon_*}$ and $\bar{\chi}_{\varepsilon_*}$. Since we used a different scale for the rates for the exact and the heuristic case, we can not compare these values directly.

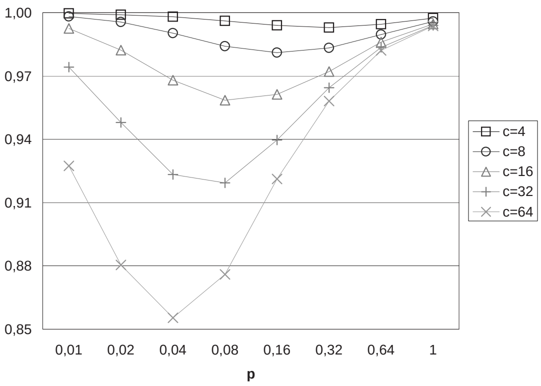


Figure 3.39: Traveling salesman problem – average time rates $\bar{\chi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^6]$

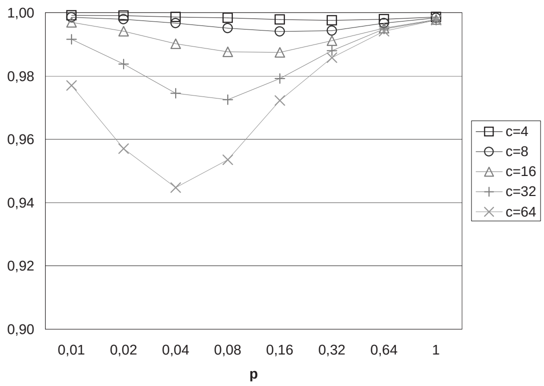


Figure 3.40: Assignment problem – average cost rates $\bar{\chi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\};\ T = 10^6]$

Dependence of ε_* on the Problem Size n

For the dependence on the problem size n we made another 10^6 runs where we computed solution pairs for the different problem sizes $n \in \{20, 40, 60, 80, 100\}$. Table 3.6 shows the decrease of the optimal penalty parameter ε_* for growing problem size n.

		n				
		20	40	60	80	100
TSP	$\varepsilon_*(n)$	0.53	0.35	0.25	0.25	0.20
	$\bar{\chi}_{\varepsilon_*}(n)$	0.903	0.958	0.975	0.983	0.987
ASP	$\varepsilon_*(n)$	0.68	0.43	0.42	0.38	0.38
	$\bar{\chi}_{\varepsilon_*}(n)$	0.960	0.989	0.994	0.996	0.997

Table 3.6: The optimal penalty parameters $\varepsilon_*(n)$ and the average time/cost rates $\bar{\chi}_{\varepsilon_*}(n)$ for the traveling salesman problem (TSP) and the assignment problem (ASP); $[p = 0.1,\ c = 32;\ I_\varepsilon = \{0.025, 0.05, \ldots, 1.5\};\ T = 10^6]$

And we see again that the rates $\bar{\chi}_{\varepsilon_*}(n)$ increase for growing problem size n. But for all tested values of n the penalty method with the 'right' penalty parameter ε_* leads on average to better results than taking a pair of two independently generated locally optimal solutions. However the benefit is rather small for larger problem sizes n.

3.4.2 The Mutual Penalty Method – Relative Improvement by Heuristic ε-Penalty Solution Pairs

Here we analyze the case that we are allowed to choose between the two solutions from a heuristic ε-penalty solution pair $\{\check{S}_{1(\varepsilon)}, \check{S}_{2(\varepsilon)}\}$ generated with the mutual penalty method. A pair of independently generated, locally optimal solutions $\{\check{S}_{l1}, \check{S}_{l2}\}$ is used as reference.

3.4.2.1 The Influence of the Penalty Parameter ε

We repeat the experiments from Subsection 3.4.1 with the mutual penalty method. Again we calculate mean values over several runs. The average rates $\bar{\chi}^m_{\varepsilon_i}$ over the whole set of instances are

$$\bar{\chi}^m_{\varepsilon_i} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(\check{S}^{(t)}_{1(\varepsilon_i)}), \widehat{w}^{(t)}(\check{S}^{(t)}_{2(\varepsilon_i)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}), \widehat{w}^{(t)}(\check{S}^{(t)}_{l2})\right)} \quad \text{for } i = 1, \ldots, N .$$

The following figures show typical results of the computer experiments. For every tested combination of the model parameters we made $T = 10^6$ independent runs. We computed $N = 30$ solution pairs for the penalty parameters $\varepsilon_1 = 0.025, \varepsilon_2 = 0.050, \ldots, \varepsilon_{30} = 0.750$.

Again Figures 3.41 and 3.42 show the expected qualitative behavior. The average rates $\bar{\chi}^m_{\varepsilon}$ are unimodal in ε. First they decrease, then they increase again. And again there exists for every tested combination of perturbation parameters p and c an optimal penalty parameter ε^m_* such that the corresponding average rate $\bar{\chi}^m_{\varepsilon_*}$ is minimal and smaller than 1.

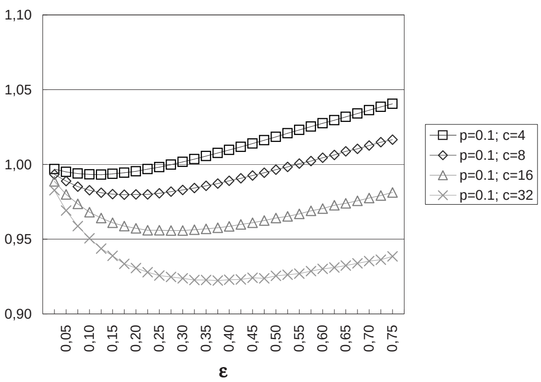


Figure 3.41: Traveling salesman problem – average time rates $\bar{\chi}^m_{\varepsilon}$ for different ε; $[n = 25;\ T = 10^6]$

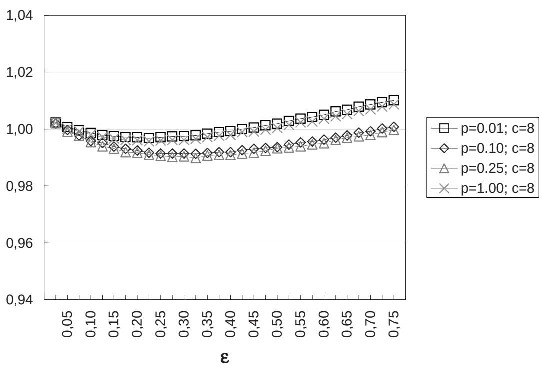


Figure 3.42: Assignment problem – average cost rates $\bar{\chi}^m_{\varepsilon}$ for different ε; $[n = 25;\ T = 10^6]$

3.4.2.2 The Optimal Penalty Parameter ε_*^m

The results of focusing the view on the optimal penalty parameter $\varepsilon_*^m(p,c,n)$ and the average rate $\bar{\chi}_{\varepsilon_*}^m(p,c,n)$ are shown in Figures 3.43 to 3.46 and Table 3.7. The qualitative behavior is identical to the case with the penalty method in Subsection 3.4.1.2.

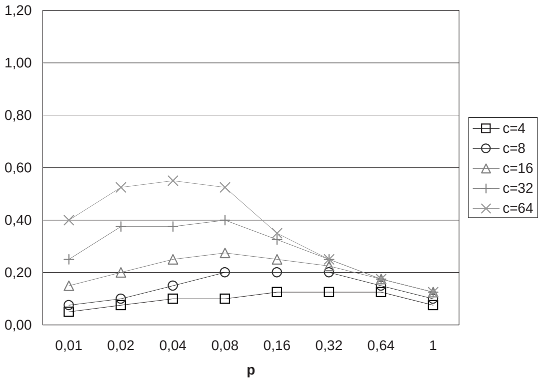


Figure 3.43: Traveling salesman problem – the optimal penalty parameters $\varepsilon_*^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.5\};\ T = 10^6]$

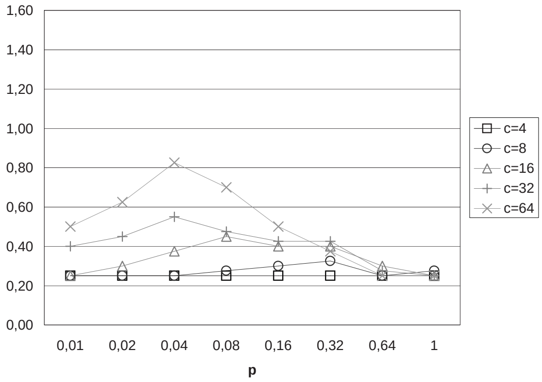


Figure 3.44: Assignment problem – the optimal penalty parameters $\varepsilon_*^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.5\};\ T = 10^6]$

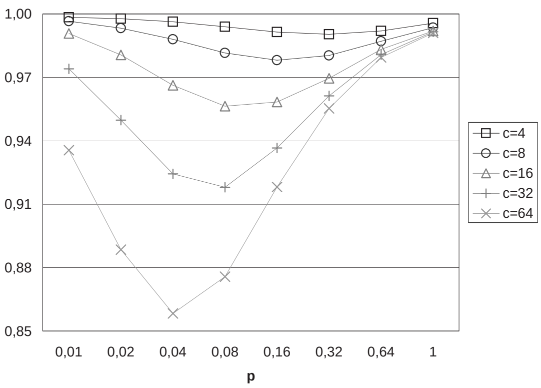


Figure 3.45: Traveling salesman problem – average time rates $\bar{\chi}_{\varepsilon_*}^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.5\};\ T = 10^6]$

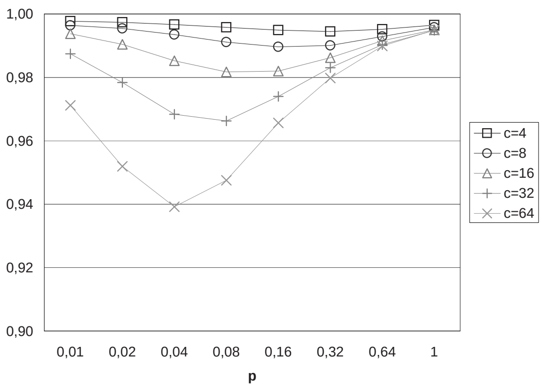


Figure 3.46: Assignment problem – average cost rates $\bar{\chi}_{\varepsilon_*}^m(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.5\};\ T = 10^6]$

		n				
		20	40	60	80	100
TSP	$\varepsilon_*^m(n)$	0.53	0.35	0.25	0.25	0.20
	$\bar{\chi}_{\varepsilon_*}^m(n)$	0.903	0.959	0.976	0.983	0.987
ASP	$\varepsilon_*^m(n)$	0.53	0.35	0.33	0.35	0.40
	$\bar{\chi}_{\varepsilon_*}^m(n)$	0.954	0.985	0.992	0.994	0.995

Table 3.7: The optimal penalty parameters $\varepsilon_*^m(n)$ and the average time/cost rates $\bar{\chi}_{\varepsilon_*}^m(n)$ for the traveling salesman problem (TSP) and the assignment problem (ASP); $[p = 0.1,\ c = 32;\ I_\varepsilon = \{0.025, 0.05, \ldots, 1.5\};\ T = 10^6]$

3.4.3 Concluding Comparison

In all of our experiments with the heuristic approaches it is advantageous to choose from two solutions generated with a penalty method. The average rates $\bar{\chi}_\varepsilon^{(m)}$ were not as impressing as in the case with exact algorithms. But if the 'right' penalty parameter ε is chosen, the heuristic penalty methods lead on average to better results than simply taking a pair of two independently generated, locally optimal solutions.

The optimal choice of the penalty parameter ε again depends on the problem, the problem size n and the simulation parameters p and c. The dependencies are identical to the case with exact algorithms (see 3.3.3).

Also as in the case with exact algorithms, the mutual approach did not show significant advantage compared to the normal penalty approach.

3.5 Stability and Variation of the Results

We conclude the experimental investigation with a brief statistical analysis of our simulation results. We will show that the average rates $\bar{\varphi}_\varepsilon$ (or $\bar{\chi}_\varepsilon$ and $\bar{\chi}_\varepsilon^m$ with heuristics) differ from the actual expected values by less than 1% with probability 98% (see [Ros 2002]). For the mutual penalty method with exact algorithms, where we did only $T = 10^4$ simulation runs, we can guarantee only maximal errors of less than 3% (with probability 98%).

Let X^t with $t = 1, \ldots, T$ be i.i.d. random variables with expected value $E[X^t]$, variance $\sigma_{X^t}^2 = Var(X^t)$ and sample mean $\bar{X} \equiv \sum_{t=1}^T \frac{X^t}{T}$. To determine the quality of $\bar{X}$ as an estimator of $E[X^t]$, we consider its mean square error, the expected value of the squared difference between $\bar{X}$ and $E[X^t]$:

$$\underbrace{E[(\bar{X} - E[X^t])^2] = Var(\bar{X})}_{\text{(since } E[\bar{X}] = E[X^t])} = Var\left(\frac{1}{T}\sum_{t=1}^{T} X^t\right) = \frac{1}{T^2}\sum_{t=1}^{T} Var(X^t) = \frac{\sigma^2_{X^t}}{T} .$$

Thus, $\bar{X}$, the sample mean of the T data values $X^1, \ldots, X^T$ is a random variable with mean $E[X^t]$ and variance $\sigma^2_{X^t}/T$. It follows that $\bar{X}$ is a good estimator of $E[X^t]$ when its standard deviation $\sigma_{\bar{X}} = \sigma_{X^t}/\sqrt{T}$ is small.

With a large T we can apply the central limit theorem to assert that

$$\frac{\bar{X} - E[X^t]}{\sigma_{X^t}/\sqrt{T}}$$

is approximately distributed like a standard normal random variable Z. Thus we have

$$P\left\{|\bar{X} - E[X^t]| > c\frac{\sigma_{X^t}}{\sqrt{T}}\right\} \approx P\{|Z| > c\} = 2[1 - \Phi(c)] ,$$

with Φ being the standard normal distribution function. Since the variance $\sigma^2_{X^t}$ is not known, we can not use the value $\sigma^2_{X^t}/T$ directly as an indication of how well the sample mean of T data values estimates the expected value. Therefore, we also need to estimate the variance

$$\sigma^2_{X^t} \approx s^2_{X^t} := \sum_{t=1}^{T} \frac{(X^t - \bar{X})^2}{T-1} = \frac{\left(\sum_{t=1}^{T} (X^t)^2\right) - T \cdot (\bar{X}^2)}{T-1} .$$

Now we are ready to evaluate our results statistically. For the penalty method with an exact algorithm let

$$\begin{aligned} X^t_\varepsilon &:= \min\{\widehat{w}^{(t)}(S^{(t)}_0), \widehat{w}^{(t)}(S^{(t)}_\varepsilon)\} \\ \text{and} \quad Y^t &:= \widehat{w}^{(t)}(S^{(t)}_0) \qquad \text{with} \quad t = 1, \ldots, T. \end{aligned}$$

From $\Phi(2.33) = 0.9901$ it follows that

$$\begin{aligned} &P\{|\bar{X}_\varepsilon - E[X^t_\varepsilon]| < 2.33 \cdot s_{X^t_\varepsilon}/\sqrt{T}\} \approx 0.98 \\ \text{and} \quad &P\{|\bar{Y} - E[Y^t]| < 2.33 \cdot s_{Y^t}/\sqrt{T}\} \approx 0.98 . \end{aligned}$$

With $\bar{\varphi}_\varepsilon = \bar{X}_\varepsilon/\bar{Y}$ and $\varphi_\varepsilon = E[X^t_\varepsilon]/E[Y^t]$ it follows that

$$\boldsymbol{P\left\{\bar{\varphi}_\varepsilon - \delta^-_\varepsilon < \varphi_\varepsilon < \bar{\varphi}_\varepsilon + \delta^+_\varepsilon\right\} \approx 0.98}$$

with the maximal deviations

$$\delta^-_\varepsilon = \bar{\varphi}_\varepsilon - \frac{\bar{X}_\varepsilon - 2.33 \cdot s_{X^t_\varepsilon}/\sqrt{T}}{\bar{Y} + 2.33 \cdot s_{Y^t}/\sqrt{T}} \quad \text{and} \quad \delta^+_\varepsilon = \frac{\bar{X}_\varepsilon + 2.33 \cdot s_{X^t_\varepsilon}/\sqrt{T}}{\bar{Y} - 2.33 \cdot s_{Y^t}/\sqrt{T}} - \bar{\varphi}_\varepsilon .$$

Since $\delta_\varepsilon^+ > 0$ and $\delta_\varepsilon^- > 0$ and

$$\delta_\varepsilon^+ - \delta_\varepsilon^- = \ldots = \frac{2 \cdot (2.33)^2 s_{Y^t}(\bar{Y} s_{X_\varepsilon^t} + \bar{X} s_{Y^t})}{\bar{Y}(\bar{Y}^2 T - (2.33 \cdot s_{Y^t})^2)} > 0 \ ,$$

we know that

$$\boldsymbol{P\{|\varphi_\varepsilon - \bar{\varphi}_\varepsilon| < \delta_\varepsilon^+\} \geq P\{\bar{\varphi}_\varepsilon - \delta_\varepsilon^- < \varphi_\varepsilon < \bar{\varphi}_\varepsilon + \delta_\varepsilon^+\} \approx 0.98} \ .$$

So we can restrict our analysis to the maximal deviations $\delta_\varepsilon := \delta_\varepsilon^+$ towards the top.

Figures 3.47 and 3.48 show with 98% certainty the maximal deviations δ_ε of the average rates $\bar{\varphi}_\varepsilon$ shown in Figures 3.1 and 3.2. For small values of ε the δ_ε slightly decrease in ε. For larger ε they are nearly constant in ε.

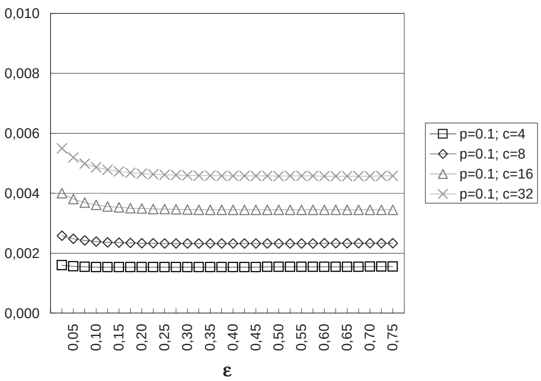


Figure 3.47: Shortest path problem – 98% certain maximal errors δ_ε of the average rates $\bar{\varphi}_\varepsilon$ shown in Figure 3.1; [$n = 25$; $T = 10^5$]

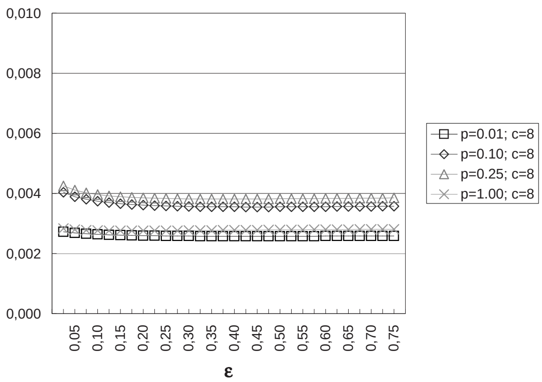


Figure 3.48: Assignment problem – 98% certain maximal errors δ_ε of the average rates $\bar{\varphi}_\varepsilon$ shown in Figure 3.2; [$n = 25$; $T = 10^5$]

Furthermore, the deviations δ_ε increase in c, are unimodal in p and decrease in the problem size n. So the maximal deviations δ_ε are approximately largest for these parameter combinations of p and c that result in small average rates $\bar{\varphi}_\varepsilon$.

Figures 3.49 and 3.50 show the maximal deviations $\delta_{\varepsilon_*}(p, c)$ for the optimal average rates $\bar{\varphi}_{\varepsilon_*}(p, c)$ (see Figures 3.9 and 3.10) shown in Subsection 3.3.1.2. Again, the maximal deviations $\delta_{\varepsilon_*}(p, c)$ are significantly smaller than 1% for each tested combination of the simulation parameters p and c.

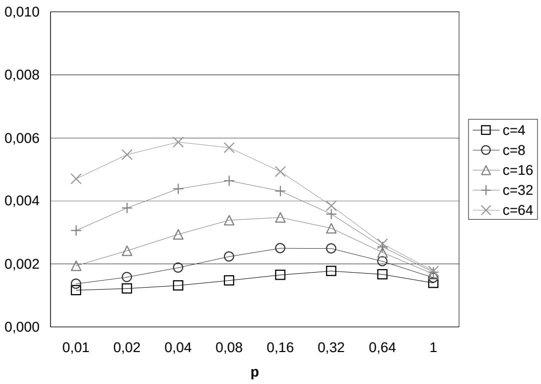


Figure 3.49: Shortest path problem – 98% certain maximal error $\delta_{\varepsilon_*}(p, c)$.

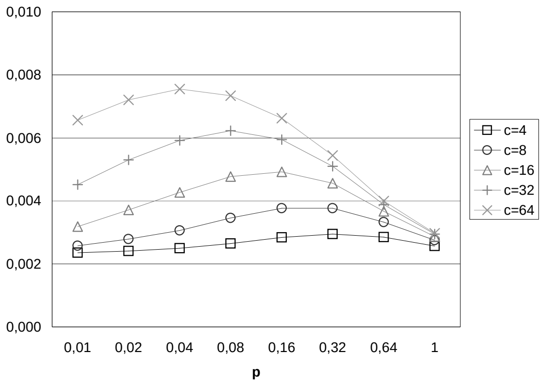


Figure 3.50: Assignment problem – 98% certain maximal error $\delta_{\varepsilon_*}(p, c)$.

For the other methods tested we have to redefine X_ε^t and Y^t appropriately, but then we can continue the calculation in the same way.

1.) mutual penalty method with exact algorithms
$$X_\varepsilon^t := \min\,(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}), \widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)})) \quad \longrightarrow \quad \delta_\varepsilon^m$$
$$Y^t := \widehat{w}^{(t)}(S_0^{(t)})$$

2.) penalty method with heuristics
$$X_\varepsilon^t := \min\,(\widehat{w}^{(t)}(\check{S}_{l1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{\varepsilon}^{(t)})) \quad \longrightarrow \quad \check{\delta}_\varepsilon$$
$$Y^t := \min\,(\widehat{w}^{(t)}(\check{S}_{l1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{l2}^{(t)}))$$

3.) mutual penalty method with heuristics
$$X_\varepsilon^t := \min\,(\widehat{w}^{(t)}(\check{S}_{1(\varepsilon)}^{(t)}), \widehat{w}^{(t)}(\check{S}_{2(\varepsilon)}^{(t)})) \quad \longrightarrow \quad \check{\delta}_\varepsilon^m$$
$$Y^t := \min\,(\widehat{w}^{(t)}(\check{S}_{l1}^{(t)}), \widehat{w}^{(t)}(\check{S}_{l2}^{(t)}))$$

Table 3.8 gives an overall view of the largest maximal errors δ_ε we observed in all the experiments. In each cell it shows the largest value δ_ε we observed for all tested parameters ε, p, c, n in the specific experiment. The complete statistical results of the single experiments are given in Appendix B.

We can conclude that the presented simulation results for the penalty method with exact algorithms are sufficiently accurate. We can be 98% certain that all presented average rates $\bar{\varphi}_\varepsilon$ differ from the exact expected values φ_ε by less than 1% (see [Ros 2002]). For the mutual penalty method we get only ±3%. The penalty methods with heuristics were statistically analyzed in the same way. With $T = 10^6$ simulation runs the maximal errors $\check{\delta}_\varepsilon$ and $\check{\delta}_\varepsilon^m$ are significantly smaller than 1%. Therefore, the significance of the results is just as high.

Subsection 3.3.1		T	$\delta_\varepsilon[\bar{\varphi}_\varepsilon]$	$\delta_{\varepsilon*}[\bar{\varphi}_{\varepsilon*}(p,c)]$	$\delta_{\varepsilon*}[\bar{\varphi}_{\varepsilon*}(n)]$
penalty method	SP	10^5	0.0055	0.0059	0.0083
with exact algorithms	ASP	10^5	0.0081	0.0076	0.0086
Subsection 3.3.2		T	$\delta^m_\varepsilon[\bar{\varphi}^m_\varepsilon]$	$\delta^m_{\varepsilon*}[\bar{\varphi}^m_{\varepsilon*}(p,c)]$	$\delta^m_{\varepsilon*}[\bar{\varphi}^m_{\varepsilon*}(n)]$
mutual penalty method	SP	10^4	0.0175	0.0185	0.0270
with exact algorithms	ASP	10^4	0.0256	0.0242	0.0277
Subsection 3.4.1		T	$\check{\delta}_\varepsilon[\bar{\chi}_\varepsilon]$	$\check{\delta}_{\varepsilon*}[\bar{\chi}_{\varepsilon*}(p,c)]$	$\check{\delta}_{\varepsilon*}[\bar{\chi}_{\varepsilon*}(n)]$
penalty method	TSP	10^6	0.0022	0.0026	0.0032
with heuristic algorithms	ASP	10^6	0.0024	0.0031	0.0035
Subsection 3.4.2		T	$\check{\delta}^m_\varepsilon[\bar{\chi}^m_\varepsilon]$	$\check{\delta}^m_{\varepsilon*}[\bar{\chi}^m_{\varepsilon*}(p,c)]$	$\check{\delta}^m_{\varepsilon*}[\bar{\chi}^m_{\varepsilon*}(n)]$
mutual penalty method	TSP	10^6	0.0022	0.0026	0.0032
with heuristic algorithms	ASP	10^6	0.0024	0.0031	0.0035

Table 3.8: The largest 98% certain maximal errors of the experiments from Sections 3.3 - 3.4.

Chapter 4

Miscellaneous Specializations and Generalizations

In this chapter we briefly discuss the following additional topics that we did not take into account so far:

4.1 Sensitivity of the Optimal Penalty Parameter ε_*

4.2 Other Perturbation Models

4.3 Specific Problem Instances

4.4 The Optimal Solution of the Perturbed Problem

4.5 The Expectation of the Quotient versus the Quotient of the Expectations

4.6 Connection to Multi-Criteria Optimization

4.1 Sensitivity of the Optimal Penalty Parameter ε_*

The experiments we reported in Chapter 3 showed that there exists an optimal penalty parameter ε_* (or ε_*^m). We observed that this optimal penalty parameter depends on the specific situation. There is not one optimal choice for all cases.

However, the Figures 3.1 and 3.2 from page 30 and Figures 3.11 and 3.12 from page 36 lead to the assumption that **for the penalty methods with exact algorithms it is better to overestimate than to underestimate the optimal penalty parameter if it is unknown**. The average rate $\bar{\varphi}_\varepsilon$ (or $\bar{\varphi}_\varepsilon^m$) decreases significantly faster for increasing $\varepsilon < \varepsilon_*$ than it increases again for $\varepsilon > \varepsilon_*$.

Figures 4.1 and 4.2 show the loss of quality

$$\triangle\bar{\varphi}_{0.1} := \bar{\varphi}_{\varepsilon_*+0.1} - \bar{\varphi}_{\varepsilon_*}$$

by overestimating the optimal penalty parameter ε_* by 0.1. In all cases the loss is smaller than 0.003. For the mutual penalty method the upper bound of the loss is a bit larger, but still small (< 0.006, see Figures B.54 and B.74 in Appendix B).

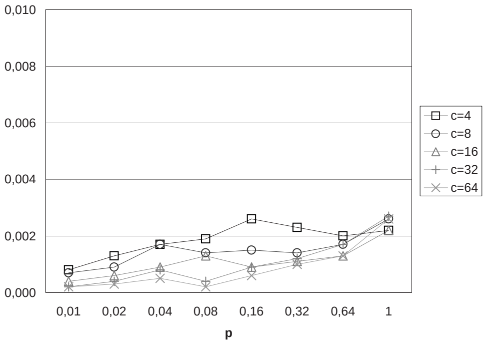


Figure 4.1: Shortest path problem – loss of quality $\triangle\bar{\varphi}_{0.1}(p,c)$ by overestimating $\varepsilon_*(p,c)$ by 0.1; [penalty method; $n = 25$; $T = 10^5$]

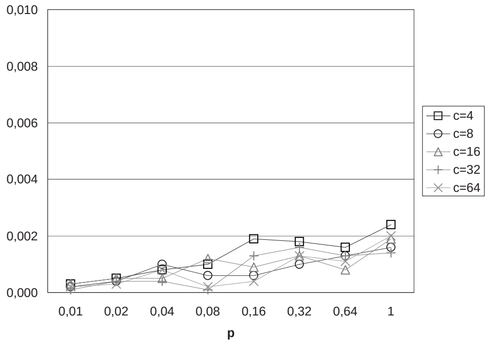


Figure 4.2: Assignment problem – loss of quality $\triangle\bar{\varphi}_{0.1}(p,c)$ by overestimating $\varepsilon_*(p,c)$ by 0.1; [penalty method; $n = 25$; $T = 10^5$]

For the penalty method with heuristics we **do not** observe that it is better to overestimate than to underestimate ε_* (see Figures 3.33 and 3.34 from page 47 and Figures 3.41 and 3.42 from page 50). Anyhow, the loss of quality

$$\triangle\bar{\chi}_{0.1} := \bar{\chi}_{\varepsilon_*+0.1} - \bar{\chi}_{\varepsilon_*}$$

by overestimating the optimal penalty parameter ε_* by 0.1 is still quite small (see Figures 4.3 and 4.4).

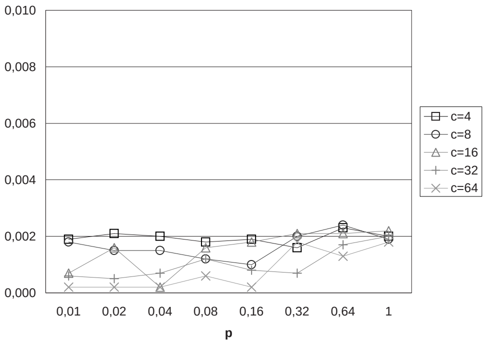


Figure 4.3: Traveling salesman problem – loss of quality $\triangle\bar{\chi}_{0.1}(p,c)$ by overestimating $\varepsilon_*(p,c)$ by 0.1; [penalty method with heuristics; $n = 25$; $T = 10^6$]

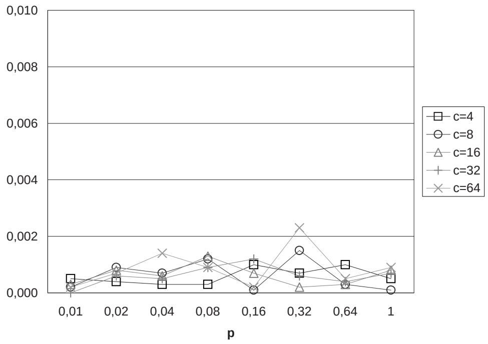


Figure 4.4: Assignment problem – loss of quality $\triangle\bar{\chi}_{0.1}(p,c)$ by overestimating $\varepsilon_*(p,c)$ by 0.1; [penalty method with heuristics; $n = 25$; $T = 10^6$]

4.2 Other Perturbation Models

All results of our experiments are based on the "*standard perturbation model*" $(p;1,c)$ (see Definition M1 on page 26), which is a generalization of the perturbation model introduced by Schwarz (see [Sch 2003]). In this section we check the validity of our results for a set of different perturbation models.

With the standard model the edge weights always increase when they get perturbed. Thus a perturbation always means a change for the worse. A first modification changes this property and results in model $(p;\frac{1}{c},c)$.

Definition M2 ($(p;\frac{1}{c},c)$-model)
Given is a $\sum$-type problem $P=(E,F,w)$. We simulate the corresponding perturbed problem by defining a new instance of weights

$$\widehat{w}(e) := \begin{cases} \lambda_c(e)\cdot w(e) & : \text{ with probability } p \\ w(e) & : \text{ with probability } 1-p \end{cases}$$

independently for all elements e. Here the $\lambda_c(e)$ are independent random numbers, uniformly distributed in the interval $[\mathbf{\frac{1}{c}}, \boldsymbol{c}] \subset \mathbb{R}$, $c>1$.

Both the $(p;1,c)$- and the $(p;\frac{1}{c},c)$-model perturb the weights independently and identically distributed for all elements $e \in \mathbb{E}$. The next model perturbs the weights $w(e)$ of the elements $e \in E$ in dependence of their weight $w(e)$. It tends to perturb "attractive" elements with a small weight $w(e)$ more likely than "unattractive" elements with a large weight $w(e)$. A similar effect would also be achieved by using additive perturbations instead of multiplicative perturbations, for example in our standard model $(p;1,c)$.

Definition M3 ($(p:=1-w(e);1,c)$-model)
Given is a $\sum$-type problem $P=(E,F,w)$ with $0\le w(e)\le 1$ for all $e\in E$. We simulate the corresponding perturbed problem by defining a new instance of weights

$$\widehat{w}(e) := \begin{cases} \lambda_c(e)\cdot w(e) & : \text{ with probability } p := 1-w(e) \\ w(e) & : \text{ with probability } 1-p \end{cases}$$

independently for all elements e. The $\lambda_c(e)$ are independent random numbers, uniformly distributed in the interval $[\mathbf{1}, \boldsymbol{c}] \subset \mathbb{R}$.

Furthermore we test an unbiased perturbation model. Here the expectation $\mathbb{E}\left[\widehat{w}(e)\right]$ of the perturbed weight $\widehat{w}(e)$ is equal to its original weight $w(e)$.

Definition M4 ((($p := \frac{1}{c+1}; 1, c$), ($1 - p; \frac{1}{c}, 1$))-model)
Given is a $\sum$-type problem $P = (E, F, w)$. We simulate the corresponding perturbed problem by defining a new instance of weights

$$\widehat{w}(e) := \begin{cases} \lambda_{c1}(e) \cdot w(e) & : \text{ with probability } p := \frac{1}{c+1} \\ \lambda_{c2}(e) \cdot w(e) & : \text{ with probability } 1 - p \end{cases}$$

independently for all elements e. The $\lambda_{c1}(e)$ are independent random numbers, uniformly distributed in the interval $[\mathbf{1}, \boldsymbol{c}] \subset \mathbb{R}$. The $\lambda_{c2}(e)$ are independent random numbers, uniformly distributed in the interval $[\frac{\mathbf{1}}{\boldsymbol{c}}, \mathbf{1}] \subset \mathbb{R}$.

Finally, we test a model where the perturbation factors $\lambda_c(e)$ are not uniformly distributed, but one sided normally distributed, based on $N(1, \sigma^2)$ with expectation $\mu = 1$ and variation σ^2 (see Figure 4.5 on page 62). The density function of this one-sided normal distribution $\boldsymbol{OSN(1, c^2)}$ is given by

$$f(x) := \begin{cases} \frac{2}{\sqrt{2\pi c^2}} \cdot e^{-\frac{(x-1)^2}{2c^2}} & : \quad x \geq 1 \\ 0 & : \quad x < 1. \end{cases}$$

Definition M5 ($\boldsymbol{OSN(1, c^2)}$-model)
Consider one of the presented $\sum$-type problems $P = (E, F, w)$. We simulate the corresponding perturbed problem by defining a new instance of weights

$$\widehat{w}(e) := \lambda_c(e) \cdot w(e)$$

independently for all elements e. The $\lambda_c(e)$ are independent random numbers, one-sided normally distributed $OSN(1, c^2)$.

For each of the four additional perturbation models M2 to M5 we performed the whole set of experiments (see Table 4.1 on page 62). The results are shown in detail in Appendices C-F. They completely support the main observations we made for our basic perturbation model:

- The average rates $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ are always unimodal in ε. It always exists an optimal penalty parameter $\varepsilon_*^{(m)}$ such that the appropriate average rate $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ is minimal with respect to ε.
- The optimal penalty parameter $\varepsilon_*^{(m)}$ depends on the problem parameters, the perturbation parameters and the method (normal or mutual penalty method).

- The optimal penalty parameter ε_*^m for the mutual penalty method is smaller than the optimal penalty parameter ε_* for the normal penalty method, when all other parameters are identical.
- For all methods, optimization problems and perturbation parameters the average rate $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ is smaller than 1 if the optimal penalty parameter $\varepsilon_*^{(m)}$ is used.
- The mutual penalty method reaches as good results as the normal penalty method, but is not better.

For the dependence of the optimal penalty parameter $\varepsilon_*^{(m)}$ on the perturbation parameters we observe that:

- For the models M3 to M5 the optimal penalty parameter $\varepsilon_*^{(m)}$ increases for increasing variation of the random perturbation factor $Var\Big(\frac{\widehat{w}(e)}{w(e)}\Big)$.
- But for our standard model $(p; 1, c)$ with

$$Var\Big(\frac{\widehat{w}(e)}{w(e)}\Big) = (c-1)^2\Big(\frac{p}{3} - \frac{p^2}{4}\Big)$$

  the above observation does not hold. Here the characteristics, that also hold for the model $M2$, are given in Section 3.3.3 (p. 39).

For the dependence of the average rates $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ on the perturbation parameters we observe that:

- The resulting average rate does not decrease with increasing expectation of the random perturbation factor $\mathbb{E}\left[\frac{\widehat{w}(e)}{w(e)}\right]$. A good indicator for this is the unbiased model $M4$. It produces the best (smallest) average rates of all models, although it holds

$$\mathbb{E}\left[\frac{\widehat{w}(e)}{w(e)}\right] = 1$$

  for all edges $e \in E$.
- The resulting average rate $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ does not decrease with increasing variation of the random perturbation factor $Var\Big(\frac{\widehat{w}(e)}{w(e)}\Big)$. For example, the perturbation factor of model $M5$ has a large variation and the appropriate average rate is worst (highest) in comparison to the other models.

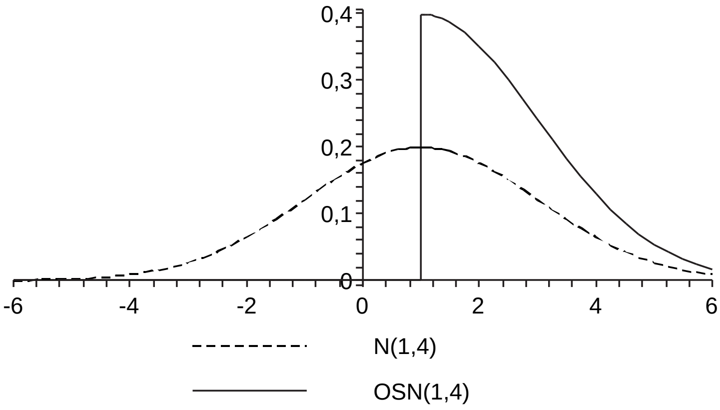


Figure 4.5: Density functions of the normal distribution $N(1, \sigma^2)$ and the corresponding one-sided normal distribution $OSN(1, c^2)$.

penalty method with exact algorithms ($T = 10^5$)	$\bar{\varphi}(\varepsilon)$	ε_*,$\bar{\varphi}_{\varepsilon_*}$
mutual penalty method with exact algorithms ($T = 10^4$)	$\bar{\varphi}^m(\varepsilon)$	ε^m_*,$\bar{\varphi}^m_{\varepsilon_*}$
SP,ASP with $n = 25$	$I_\varepsilon = \{0.025, 0.050, \ldots, 0.750\}$	$I_\varepsilon = \{0.025, 0.050, \ldots, 2.500\}$
penalty method with heuristics ($T = 10^6$)	$\bar{\chi}(\varepsilon)$	ε_*,$\bar{\chi}_{\varepsilon_*}$
mutual penalty method with heuristics ($T = 10^6$)	$\bar{\chi}^m(\varepsilon)$	ε^m_*,$\bar{\chi}^m_{\varepsilon_*}$
ASP,TSP with $n = 25$	$I_\varepsilon = \{0.025, 0.050, \ldots, 0.750\}$	$I_\varepsilon = \{0.025, 0.050, \ldots, 2.500\}$
M2	$p = 0.1; c = 4, 8, 16, 32;$ and $p = 0.01, 0.1, 0.25, 1.00; c = 8$	$p = 0.01, 0.02, 0.04, 0.08, 0.16, 0.32,$ $0.64, 1.00;$ $c = 4, 8, 16, 32, 64$
M3,M4	$c = 4, 8, 16, 32;$	$c = 2, 4, 8, 16, 32, 64$
M5	$c = 1, 2, 4, 8;$	$c = 1, 2, 4, 8, 16, 32$

Table 4.1: Overview of the experiments with different perturbation models.

4.3 Specific Problem Instances

In Chapter 3 we always analyzed a situation where only the class of a problem is given, but in some situations a specific fixed instance of a problem may be given. Then the planner is interested in the behavior of this specific problem.

4.3.1 Specific Perturbation Instances

If in a specific problem instance we look at one specific instance of perturbations, then completely different behavior may occur depending on the problem instance and the perturbations. Especially the following may happen: with respect to the perturbed weights $\widehat{w}$ the original optimal solution S_0 is better than some alternatives S_{ε_i} with $i \in I_1$ ($\varphi_{\varepsilon_i} = 1$ for all $i \in I_1$) and worse than some other alternatives S_{ε_i} with $i \in I_2$ ($\varphi_{\varepsilon_i} < 1$ for all $i \in I_2$). In some cases one of the two index sets is the empty set: $I_1 = \emptyset$ or $I_2 = \emptyset$. Figures 4.6 and 4.7 show two possible situations. Other examples can look quite different.

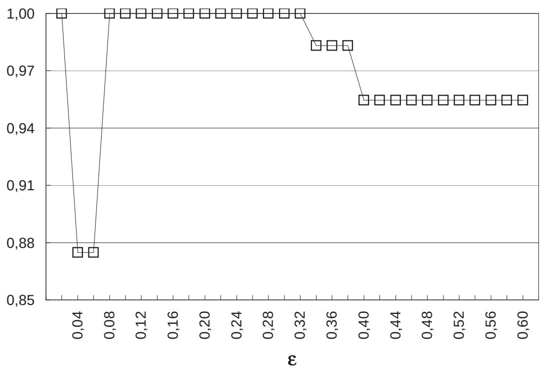


Figure 4.6: Shortest path problem – time rates $\varphi_{\varepsilon=0.02}, \varphi_{\varepsilon=0.04}, \ldots, \varphi_{\varepsilon=0.60}$ for one randomly generated 'shortest path problem'–instance with one randomly generated instance of perturbed weights; $[n = 25, p = 0.25, c = 8]$

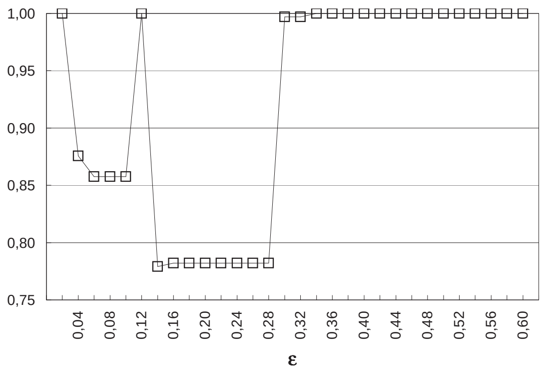


Figure 4.7: Assignment problem – cost rates $\varphi_{\varepsilon=0.02}, \varphi_{\varepsilon=0.04}, \ldots, \varphi_{\varepsilon=0.60}$ for one randomly generated 'assignment problem'–instance with one randomly generated instance of perturbed weights; $[n = 25, p = 0.125, c = 8]$

For every instance of an optimization problem the rates φ_ε decrease and increase in steps for growing ε. There is always a limited number of intervals $[a_i, a_{i+1}] \subset \mathbb{R}$ for which the penalty method generates the same ε-penalty solution for all $\varepsilon \in [a_i, a_{i+1}]$. So the rates φ_ε of the solution pairs $\{S_0, S_\varepsilon\}$ are equal for all $\varepsilon \in [a_i, a_{i+1}]$. Obviously these intervals depend only on the problem instance. The height of the steps depends on the perturbation instance, the perturbation parameters p, c and the tested problem instance.

4.3.2 Average Perturbation Instances

In a next step we look at the average behavior of the rates φ_{ε_i}, still for a fixed problem, but for samples of randomly generated perturbed weights $\widehat{w}$. We randomly generate a problem $P = (E, F, w)$ and calculate all the solution pairs $\{S_0, S_{\varepsilon_1}\}, \{S_0, S_{\varepsilon_2}\}, \ldots, \{S_0, S_{\varepsilon_N}\}$. Then we iteratively repeat for $t = 1, 2, \ldots, T$ the following two steps:

1. Randomly generate an instance of perturbed weights $\widehat{w}^{(t)}$.
2. Evaluate the solution pairs with respect to $\widehat{w}^{(t)}$.

Then the average rates $\widetilde{\varphi}_{\varepsilon_i}$ of the specific problem instance are:

$$\widetilde{\varphi}_{\varepsilon_i} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0), \widehat{w}^{(t)}(S_{\varepsilon_i})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_0)} \quad \text{for } i = 1, \ldots, N\,.$$

Figure 4.8 shows (for exactly the same 'shortest path problem'-instance as in Figure 4.6) the average rates for $T = 10^5$ independently generated random instances of perturbed weights. And Figure 4.9 shows (for exactly the same 'assignment-problem'-instance as in Figure 4.7) the average rates for $T = 10^5$ independently generated instances of perturbed weights, respectively.

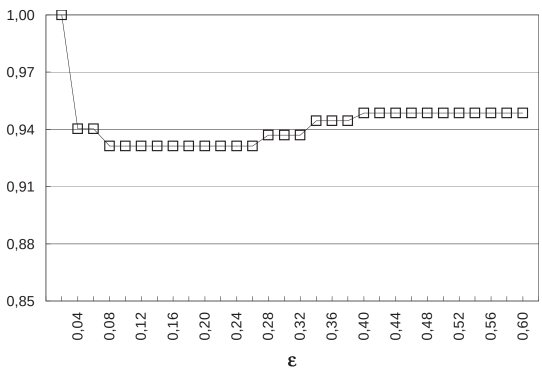


Figure 4.8: Shortest path problem – average time rates $\widetilde{\varphi}_{\varepsilon=0.02}, \widetilde{\varphi}_{\varepsilon=0.04}, \ldots, \widetilde{\varphi}_{\varepsilon=0.60}$ for one randomly generated 'shortest path problem'–instance with 10^5 randomly generated instances of perturbed weights; $[n = 25, p = 0.25, c = 8]$

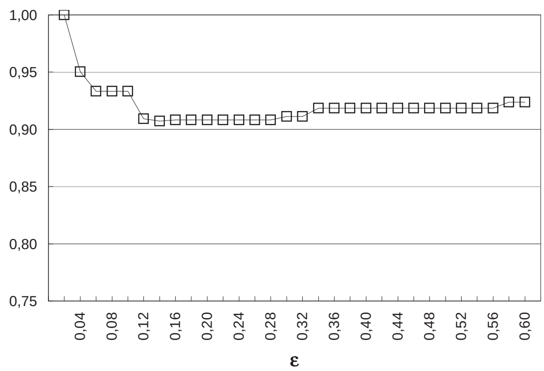


Figure 4.9: Assignment problem – average cost rates $\widetilde{\varphi}_{\varepsilon=0.02}, \widetilde{\varphi}_{\varepsilon=0.04}, \ldots, \widetilde{\varphi}_{\varepsilon=0.60}$ for one randomly generated 'assignment problem'–instance with 10^5 randomly generated instances of perturbed weights; $[n = 25, p = 0.25, c = 8]$

The average rates first decrease and then increase stepwise. There are always intervals $[a_i, a_{i+1}] \subset \mathbb{R}$ where the average rates $\widetilde{\varphi}_\varepsilon$ of the solution pairs $\{S_0, S_\varepsilon\}$ are constant for all $\varepsilon \in [a_i, a_{i+1}]$.

In a typical situation with $0 < p \leq 1$ and $c > 1$ there exists an intermediate interval of optimal penalty parameters $[\varepsilon_{*_1}, \varepsilon_{*_2}]$ such that:

$$\widetilde{\varphi}_{\varepsilon_*} < \widetilde{\varphi}_{\varepsilon} \quad \text{for all} \quad \varepsilon_* \in [\varepsilon_{*_1}, \varepsilon_{*_2}] \quad \text{and} \quad \varepsilon \notin [\varepsilon_{*_1}, \varepsilon_{*_2}] \tag{4.1}$$

$$\widetilde{\varphi}_{\varepsilon_l} \geq \widetilde{\varphi}_{\varepsilon_r} \quad \text{for all} \quad \varepsilon_l < \varepsilon_r < \varepsilon_{*_1} \tag{4.2}$$

$$\widetilde{\varphi}_{\varepsilon_l} \leq \widetilde{\varphi}_{\varepsilon_r} \quad \text{for all} \quad \varepsilon_{*_2} < \varepsilon_l < \varepsilon_r. \tag{4.3}$$

But in rare cases this property is not fulfilled. Then there are two (or more) intervals of locally optimal penalty parameters. Figures 4.10 and 4.11 show examples.

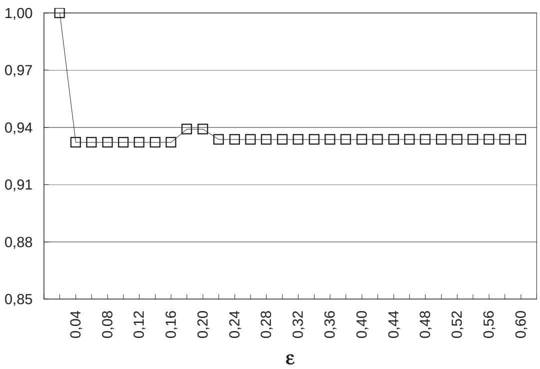


Figure 4.10: Shortest path problem – average time rates $\widetilde{\varphi}_{\varepsilon=0.02}, \widetilde{\varphi}_{\varepsilon=0.04}, \ldots, \widetilde{\varphi}_{\varepsilon=0.60}$ for one randomly generated 'shortest path problem'–instance with 10^5 randomly generated instances of perturbed weights; $[n = 25, p = 0.25, c = 8]$

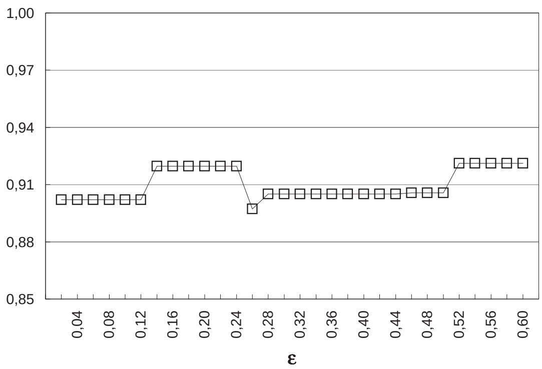


Figure 4.11: Assignment problem – average cost rates $\widetilde{\varphi}_{\varepsilon=0.02}, \widetilde{\varphi}_{\varepsilon=0.04}, \ldots, \widetilde{\varphi}_{\varepsilon=0.60}$ for one randomly generated 'assignment problem'–instance with 10^5 randomly generated instances of perturbed weights; $[n = 25, p = 0.25, c = 8]$

4.4 The Optimal Solution of the Perturbed Problem

In all our experiments it was allowed to choose from two different solutions (for example S_0 and S_ε) of a model with uncertain data. At the time of the decision the "final" $\widehat{w}$-values of the solutions S_0 and S_ε were known. The choice between S_0 and S_ε improved the situation significantly.

$$\bar{\varphi}_{\varepsilon_*} = \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_{\varepsilon_*}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(S_0^{(t)})}$$

Here we compare the minimal $\widehat{w}$-value of the two solutions with the value of the optimal solution for the perturbed problem. Let $\widehat{S}_{opt}$ be this optimal solution with respect to the true weights $\widehat{w}$. The appropriate average performance rate is given by

$$\widehat{\bar{\varphi}_{\varepsilon_*}} := \frac{\frac{1}{T}\sum_{t=1}^{T} \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_{\varepsilon_*}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T} \widehat{w}^{(t)}(\widehat{S}_{opt}^{(t)})}.$$

Whereas the "old" rates $\bar{\varphi}_{\varepsilon_*}$ were smaller or equal to 1, the new rates $\widehat{\bar{\varphi}_{\varepsilon_*}}$ have to be larger or equal to 1. It can be expected that the new rates $\widehat{\bar{\varphi}_{\varepsilon_*}}$ are even significantly larger than 1 for certain combinations of the perturbation parameters p and c.

Anyhow, it is quite interesting to compare both cases for different perturbation probabilities p and perturbation widths c. Figure 4.12 (identical to Figure 3.9 on page 34) shows how much can be saved by the choice between S_0 and S_ε in comparison to the reference case where only the original optimal solution S_0 is given. And Figure 4.13 shows how much the choice between S_0 and S_ε is worth in comparison to the true optimal solution $\widehat{S}_{opt}$. Both Figures are for the shortest path problem.

In contrast, we present the data of both cases for the assignment problem in one table (see Table 4.2). There are two entries in every cell, separated by a colon. First (in brackets), the "old" average rates $\bar{\varphi}_{\varepsilon_*}(p, c)$ with respect to the former optimal solution S_0 are given. And second, the new rates $\widehat{\bar{\varphi}_{\varepsilon_*}}(p, c)$ with respect to the optimal solution of the perturbed problem are given.

The "new" average rates $\widehat{\bar{\varphi}_{\varepsilon_*}}$ are increasing in c and unimodal in p. In the most extreme case with $p = 0.32$ and $c = 64$ the average minimal value of the solution pair is about six times as large as the average value of the new optimum. Thus it would be a huge advantage to know the "true" optimum $\widehat{S}_{opt}$. The choice

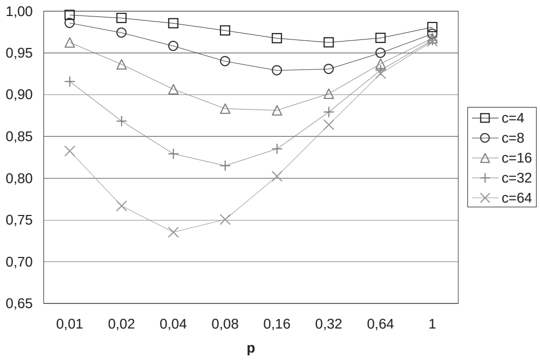


Figure 4.12: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon_*}(p,c)$ with respect to the optimal solution of the original problem; [$n = 25$; $I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\}$; $T = 10^5$]

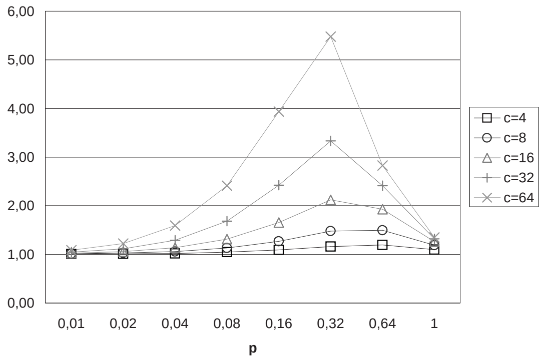


Figure 4.13: Shortest path problem – average time rates $\widehat{\bar{\varphi}_{\varepsilon_*}}(p,c)$ with respect to the optimal solution of the perturbed problem; [$n = 25$; $I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\}$; $T = 10^5$]

$p \backslash c$	4	8	16	32	64
0.01	(1.00) : 1.00	(0.99) : 1.01	(0.96) : 1.03	(0.92) : 1.05	(0.83) : 1.08
0.02	(0.99) : 1.01	(0.97) : 1.03	(0.93) : 1.06	(0.86) : 1.11	(0.75) : 1.20
0.04	(0.98) : 1.02	(0.95) : 1.06	(0.89) : 1.12	(0.80) : 1.25	(0.69) : 1.49
0.08	(0.97) : 1.04	(0.93) : 1.12	(0.86) : 1.28	(0.77) : 1.59	(0.68) : 2.21
0.16	(0.96) : 1.09	(0.91) : 1.26	(0.84) : 1.61	(0.78) : 2.34	(0.74) : 3.78
0.32	(0.95) : 1.16	(0.91) : 1.49	(0.86) : 2.17	(0.84) : 3.56	(0.82) : 6.34
0.64	(0.96) : 1.23	(0.93) : 1.66	(0.91) : 2.53	(0.90) : 4.26	(0.90) : 7.64
1.00	(0.97) : 1.12	(0.96) : 1.29	(0.96) : 1.50	(0.95) : 1.74	(0.95) : 1.95

Table 4.2: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon_*}$ with respect to the optimal solution of the original problem in contrast to the average cost rates $\widehat{\bar{\varphi}_{\varepsilon_*}}$ with respect to the optimal solution of the perturbed problem; shown in the form $(\bar{\varphi}_{\varepsilon_*}) : \widehat{\bar{\varphi}_{\varepsilon_*}}$ [penalty method; $n = 25$; $I_\varepsilon = \{0.025, 0.05, \ldots, 2.5\}$; $T = 10^5$]

from good candidate solutions is only profitable as long as this new optimum is not available or the time/cost needed to get it is too large.

In case of seldom but severe perturbations the situation is different. For example, with $p = 0.01$ and $c = 64$ the choice from two solutions S_0 and S_{ε_*} saves on average about 17% in comparison to the case where only the original optimal solution S_0 is known. And this minimal value of S_0 and S_{ε_*} is on average only about 8% larger than the value of the true optimum $\widehat{S}_{opt}$.

4.5 The Expectation of the Quotient versus the Quotient of the Expectations

In Subsection 3.1.1 we introduced our experiments. We announced to analyze the expected performance improvement by the choice between two alternatives S_{a_1} and S_{a_2} in comparison to the situation without choice. We scaled the expected minimal value of the two alternatives S_{a_1} and S_{a_2} by the expected value of the original optimal solution S_0:

$$\varphi^{(\mathbb{E}/\mathbb{E})} := \frac{\mathbb{E}\ \min\left(\widehat{w}(S_{a_1}), \widehat{w}(S_{a_2})\right)}{\mathbb{E}\ \widehat{w}(S_0)} \ .$$

Instead we could have defined the expected performance ratio also by the expectation of the rate

$$\varphi^{(\mathbb{E})} := \mathbb{E}\left[\frac{\min\left(\widehat{w}(S_{a_1}), \widehat{w}(S_{a_2})\right)}{\widehat{w}(S_0)}\right] \ .$$

The following example illustrates why the expected value of the quotient (in contrast to the ratio of the expectations) is actually not suitable for our analysis. It misrepresents the situation and can easily be misinterpreted.

Consider two identical algorithms A_X and A_Y. For a certain minimization problem they randomly result in one of n possible solutions with the values

$$X = \{x_1, x_2, \ldots, x_n\} \quad \text{and} \quad Y = \{y_1, y_2, \ldots, y_n\}$$

with

$$0 < x_i = y_i \quad \text{for all} \quad i = 1, 2, \ldots, n$$

and

$$P(X = x_i, Y = y_j) = p_{x_i y_j} = p_{x_j y_i} \quad \text{for all} \quad i = 1, 2, \ldots, n \ .$$

For all $x_i \neq y_j$ we have

$$\begin{aligned} & p_{x_i y_j} \cdot \frac{x_i}{y_j} + p_{x_j y_i} \cdot \frac{x_j}{y_i} \\ = {} & p_{x_i y_j} \cdot \left(\frac{x_i}{y_j} + \frac{x_j}{y_i}\right) > 2 \cdot p_{x_i y_j} = p_{x_i y_j} + p_{x_j y_i} \ , \end{aligned}$$

since

$$\frac{x_i}{y_j} + \frac{x_j}{y_i} = \frac{x_i y_i + x_j y_j}{y_i y_j} = \frac{x_i^2 + y_j^2}{x_i y_j} = \frac{x_i^2 - 2x_i y_j + y_j^2}{x_i y_j} + 2 = \frac{\overbrace{(x_i - y_j)^2}^{>0}}{\underbrace{x_i y_j}_{>0}} + 2 > 2 \, .$$

If there is at least one pair $i \neq j$ with $p_{x_i y_j} > 0$, then

$$\begin{aligned}
\mathbb{E}\left(\frac{X}{Y}\right) &= \sum_{i=1}^{n}\sum_{j=1}^{n} p_{x_i y_j} \cdot \frac{x_i}{y_j} \\
&= \sum_{i=1}^{n} p_{x_i y_i} \cdot \underbrace{\frac{x_i}{y_i}}_{=1} \quad + \quad \sum_{i=1}^{n}\sum_{j=i+1}^{n} p_{x_i y_j} \cdot \underbrace{\left(\frac{x_i}{y_j} + \frac{x_j}{y_i}\right)}_{>2} \\
&> \sum_{i=1}^{n}\sum_{j=1}^{n} p_{x_i y_j} = 1\,,
\end{aligned}$$

but by symmetrie of course also

$$\mathbb{E}\left(\frac{Y}{X}\right) > 1\,.$$

Thus it obviously is wrong to conclude **from $\mathbb{E}\left(\frac{X}{Y}\right) > 1$** that algorithm A_Y is better on average than algorithm A_X.

4.6 Connection to Multi-Criteria Optimization

This section establishes a connection between the penalty method and multicriteria optimization (see [Ehr 2000], [GHS 1999]). Thus we have to introduce some standard concepts. Consider a $\sum$-type problem $P = (E, F, w)$. The problem

$$\min_{S \in F} (f_1(S), f_2(S))$$

with two different objective functions f_1 and f_2 is a bi-criteria optimization problem. If there is no priority for either f_1 or f_2, then there exist in general several solutions (the "efficient frontier") from which none is better than any other with respect to both objectives. These solutions are called Pareto-optimal. More formally:

Definition 4.6.1
A solution $S_* \in F$ is called weakly (strictly) ***Pareto-optimal****, if there is no $S \in F$, $S \neq S_*$ such that $f_i(S) < f_i(S_*)$ for $i = 1, 2$ ($f_i(S) \leq f_i(S_*)$ for $i = 1, 2$).*

A solution $S_1 \in F$ ***dominates*** *a solution $S_2 \in F$, if $f_i(S_1) \leq f_i(S_2)$ for $i = 1, 2$ and $f_j(S_1) < f_j(S_2)$ for $j = 1$ or $j = 2$.*

By a **weighted sums scalarization** the bi-criteria optimization problem can be reduced to a single criteria optimization problem:

$$\min_{S \in F} f(S)$$
$$\text{with} \quad f(S) = \lambda_1 f_1(S) + \lambda_2 f_2(S) \quad \text{and} \quad \lambda_1, \lambda_2 \geq 0 \ .$$

This problem can be solved with traditional algorithms. But in some situations this method has a big disadvantage that is illustrated by the following example.

Every dot in Figure 4.14 represents one of the 10^4 shortest paths of a 25×25 grid graph evaluated by two objective functions. The horizontal axis represents

$$f_1(S) := w(S), \quad \text{the length of the path}$$

and the vertical axis represents

$$f_2(S) := w(S \cap S_0), \quad \text{the shared length with the optimal solution } S_0.$$

All Pareto-optimal solutions - the interesting alternative paths - are indicated by the symbol $\square$. All paths that can be optimal solutions of the weighted sums problem above for a certain combination of the weights $\lambda_1, \lambda_2 \geq 0$ are additionally marked by the symbol $\times$. In this example, there are several Pareto-optimal solutions that cannot be optimal for the weighted sums method, for instance the solution-triple in the middle of the figure (length ≈ 11.5, shared length ≈ 7.3). These solutions are called **non-supported non-dominated solutions**.

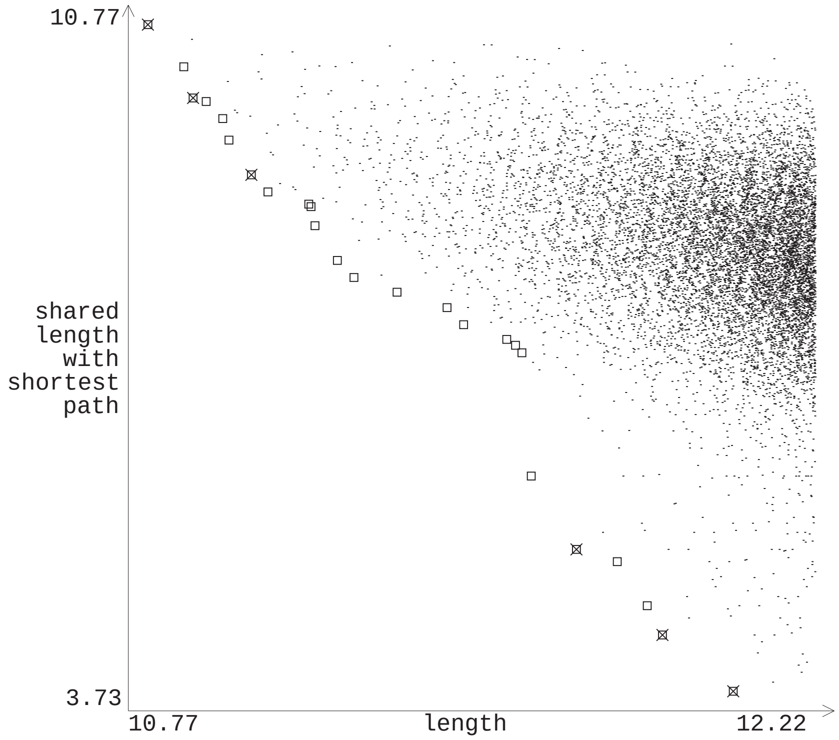


Figure 4.14: The 10^4 shortest paths of a randomly generated 25×25 grid graph sorted by their length (horizontal axis) and their shared length with the shortest path (vertical axis).

They are Pareto-optimal solutions that are not on the border of the convex hull of the solution space. But if f is a positive linear combination of f_1 and f_2, obviously only a solution that is on this border can be optimal for f.

Thus there are problems for which a weighted sums approach can reach only a part of the Pareto-optimal solutions. In critical examples this part can be very small.

The penalty method corresponds to a weighted sums approach with the two objective functions $f_1(S) = w(S)$ and $f_2(S) = w(S \cap S_0)$ and the weights $\lambda_1 := 1$ and $\lambda_2 := \varepsilon$. Thus it has the disadvantage of missing non-supported non-dominated solutions, if F is not a convex, compact set with respect to f_1 and f_2 (see [Ehr 2000]).

With this background it could be promising to have a closer look at the theory of multicriteria optimization. It might deliver algorithms that are more complex and thus need more computation time, but do not have the disadvantage of missing non-supported non-dominated solutions.

Chapter 5

Conclusions and Open Problems

5.1 Conclusions

→ In all our experiments it turned out that it is clearly advantageous to have two different solutions to choose from. In the case of seldom but severe perturbations the average benefits are largest.

→ These benefits get only realized if a good penalty parameter ε is chosen. In our experiments there always existed an optimal intermediate penalty parameter ε_* such that the benefit was maximal. The more the penalty parameter in use differed from the optimal penalty parameter, the worse the average results were.

→ Unfortunately (in view of practice), the optimal penalty parameter ε_* depends on the problem, the problem size n and the simulation parameters p and c. The following rules of thumb give a general idea of the dependencies:

The optimal penalty parameter ε_*

- monotonically decreases in the problem size n.
- monotonically increases in the perturbation width c.
- first increases slightly then decreases in the perturbation probability p. For small perturbation widths c the penalty parameter ε_* is nearly constant in p.

→ Furthermore, we conjecture that it is better to overestimate the optimal penalty parameter ε_* than to underestimate it by the same amount.

→ In our experiments the mutual penalty method reached as good results as the normal penalty method, but was not better.

→ Whereas the dependencies of the optimal penalty parameter ε_* are qualitatively equal for the normal and the mutual penalty method, the penalty parameters differ quantitatively. For the mutual approach the optimal penalty parameter ε_*^m tends to be a bit smaller than ε_*.

→ When a good penalty parameter ε is chosen, even the use of the penalty method with heuristics leads on average to better results than simply taking a pair of independently generated heuristic solutions.

→ The penalty method is easily implemented. The computation of ε-penalty solutions is as easy as the computation of the optimum in the original problem.

5.2 Open Problems

→ The unimodal behavior of the average rates $\bar{\varphi}^{(m)}/\bar{\chi}^{(m)}$ with respect to the penalty parameter ε and thus the existence of the optimal intermediate penalty parameter $\varepsilon_*^{(m)}$ is only an experimental result. It should be proved theoretically. A first investigation for smaller problems was done and will be documented in [Sam 2005b].

→ The results of our investigation of course depend on the definition of the performance ratio

$$\varphi = \frac{\mathbb{E}\ \min\left(\widehat{w}(S_{a_1}), \widehat{w}(S_{a_2})\right)}{\mathbb{E}\ \widehat{w}(S_0)} \approx \frac{\frac{1}{T}\sum_{t=1}^{T}\ \min\left(\widehat{w}^{(t)}(S_{a_1}^{(t)}), \widehat{w}^{(t)}(S_{a_2}^{(t)})\right)}{\frac{1}{T}\sum_{t=1}^{T}\ \widehat{w}^{(t)}(S_0^{(t)})}\ .$$

We showed in Section 4.5 that the expectation of the quotient

$$\mathbb{E}\left[\frac{\min\left(\widehat{w}(S_{a_1}), \widehat{w}(S_{a_2})\right)}{\widehat{w}(S_0)}\right] \approx \frac{1}{T}\sum_{t=1}^{T} \frac{\min\left(\widehat{w}^{(t)}(S_{a_1}^{(t)}), \widehat{w}^{(t)}(S_{a_2}^{(t)})\right)}{\widehat{w}^{(t)}(S_0^{(t)})}$$

is not a suitable definition for our purposes.
Another possibility would be to define the average performance ratio by the geometric mean:

$$\bar{\varphi}_G = \sqrt[T]{\prod_{t=1}^{T} \frac{\min\left(\widehat{w}^{(t)}(S_{a_1}^{(t)}), \widehat{w}^{(t)}(S_{a_2}^{(t)})\right)}{\widehat{w}^{(t)}(S_0^{(t)})}} = \frac{\sqrt[T]{\prod_{t=1}^{T}\ \min\left(\widehat{w}^{(t)}(S_{a_1}^{(t)}), \widehat{w}^{(t)}(S_{a_2}^{(t)})\right)}}{\sqrt[T]{\prod_{t=1}^{T}\ \widehat{w}^{(t)}(S_0^{(t)})}}\ .$$

Here the mean of the quotients is identical to the quotient of the means.

→ Our classes of randomly generated optimization problems and our perturbation models are quite artificial. The behavior of alternative "real-world" models that are observed from specific practical applications should be analyzed.

→ We focused on the case with two different solutions to choose from. The penalty method can also be used to generate more than one alternative. A first investigation was done by Schwarz (see [Sch 2003], p. 45).

→ Are there instances of $\sum$-type problems and combinations of parameters (p, c) for our standard perturbation model $(p; 1, c)$ such that a non-dominated non-supported solution S_{nsnd} of the corresponding bi-criteria optimization problem (see Subsection 4.6) is a better alternative than the optimal ε-penalty alternative S_{ε_*}?

→ Real-life experiments with human supervisors should be done to show the benefit of multiple choice systems in non-artificial environments. For example, high-level strategic board games seem to be ideal for such experiments (see [AS 2002],[Alt 2004]).

References

[ABS 2002] I. Althöfer, F. Berger, S. Schwarz. (2002) *Generating true alternatives with a penalty method.* Preprint, available at
`http://www.minet.uni-jena.de/Math-Net/reports/shadows/02-04report.html` .

[ABSS 2004] I. Althöfer, F. Berger, J. Sameith, S. Schwarz. (2004) *Multiple-Choice-Vorschläge für Realzeit-Entscheidungen.* OR News Nr.21, Gesellschaft für Operations Research (GOR) e.V., Bochum.

[AL 2003] E. Aarts, J.K. Lenstra (Eds.) (2003) *Local search in combinatorial optimization.* Princeton University Press, Princeton.

[Alt 2000] I. Althöfer. (2000) *Decision support systems with multiple choice structure.* In: *"Numbers, Information and Complexity."* Kluwer, Dordrecht, pp.525-540.

[Alt 2004] I. Althöfer. (2004) *Improved game play by multiple computer hints.* Theoretical Computer Science, vol.313, pp.315-324.

[AMO 1993] R.K. Ahuja, T.L. Magnanti, J.B. Orlin. (1993) *Network Flows : Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs, NJ, pp.93-157, pp.461-501.

[AS 2002] I. Althöfer, R.G. Snatzke. (2002) *Playing games with multiple choice systems.* in: J. Schaeffer, M. Müller, Y. Björnsson (Eds.) *Computers and Games.* CG 2002, Third International Conference, Revised Papers, LNCS, Edmonton, Canada.

[Ber 2000] F. Berger. (2000) *K alternative paths instead of k shortest paths.* Diploma thesis, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena.

[BEN 2004] M. Berkelaar, K. Eikland, P. Notebaert. (2004) `lp_solve` - *Open source (Mixed - Integer) Linear Programming system.* Version 5.1.0.0 dated 1 september 2004, GNU Lesser General Public Licence, available at
`http://groups.yahoo.com/group/lp_solve/` .

[Ehr 2000] M. Ehrgott. (2000) *Multicriteria Optimization.* Lecture Notes in Economics and Mathematical Systems, vol.491, Springer, Berlin.

[Epp 1998] D. Eppstein. (1998) *Finding the k shortest paths.* SIAM J. Computing, Vol.28, No.2, pp.652-673.

[GHS 1999] T. Gal, T. Hanne, T. J. Stewart (Eds.) (1999) *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications.* Kluwer, Dordrecht.

[JM 1999] V. Jiménez, A. Marzal. (1999) Website on: *Algorithms for Computing the K Shortest Paths.* Universitat Jaume I , Castellón, Spain. `http://terra.act.uji.es/REA/` .

[LSSW 1998] T.C. Lai, Y.N. Sotskov, N.Y. Sotskova, and F. Werner. (1998) *Mean flow time minimization with uncertain processing times.* Otto-von-Guericke-Universität Magdeburg, FMA, Preprint 15/98.

[MN 1998] M. Matsumoto and T. Nishimura. (1998) *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator.* ACM Trans. on Modeling and Computer Simulation Vol.8, No.1, January, pp.3-30.

[MN 2002] M. Matsumoto and T. Nishimura. (2002) *A C-program for MT19937, with initialization improved 2002/1/26.* available at `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html` .

[MS 1991] K. Manteuffel, E. Seiffart. (1991) *Lineare Optimierung.* Teubner, Leipzig, pp.139-147.

[Rei 1994] G. Reinelt. (1994) *The Traveling Salesman : Computational Solutions for TSP Applications.* Springer, Berlin.

[Ros 2002] S. M. Ross. (2002) *Simulation.* 3rd Edition, Academic Press, San Diego, pp.109-115.

[Sam 2004] J. Sameith. (2004) *On the Generation of Alternative Solutions for Discrete Optimization Problems with Uncertain Data – An Experimental Analysis of the Penalty Method.* Preprint, available at `http://www.minet.uni-jena.de/math-net/reports/shadows/04-01report.html` .

[Sam 2005a] J. Sameith. (2005) *The Penalty Method for Optimization Problems with Uncertain Data.* Submitted to Mathematical Methods of Operations Research.

[Sam 2005b] J. Sameith. (2005) *Relative Improvement by Alternative Solutions for Classes of Simple Shortest Path Problems with Uncertain Data.* Technical report in preparation. Available from autumn 2005 at
http://www.minet.uni-jena.de/math-net/reports/ .

[Sch 2003] S. Schwarz. (2003) *On the generation of multiple candidate solutions for discrete optimization problems.* Doctoral dissertation, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena.

Appendix A

Notations and Basics

A.1 Notations

A.1.1 General

Investigated Optimization Problems

- $P = (E, F, w)$ defines a $\sum$-type problem (see Definitions 1.1.1 p.5; 1.1.2.1 p.6; 1.1.2.2 p.6 and 1.1.2.3 p.7).

- $\widehat{w}$ is a perturbed instance of the weight function w of problem P. The original weight $w(e)$ of each edge e gets perturbed by a random variable Δ_e:

$$\widehat{w}(e) := \Delta_e(w(e)) \quad \text{for all} \quad e \in E.$$

  See Definitions M1 p.26; M2 p.59; M3 p.59; M4 p.60 and M5 p.60.

Solutions derived by exact algorithms

- S_0 denotes a globally optimal solution of P.

- S_ε denotes an ε-penalty solution of P derived from the penalty method with respect to solution S_0 (see Definition 2.2.1 p.11).

- $\{S_{1(\varepsilon)}, S_{2(\varepsilon)}\}$ denotes an ε-penalty solution-pair of P derived from the mutual penalty method (see Definition 2.3.1 p.15).

Solutions derived by heuristic algorithms

- $\check{S}_{l1}$ and $\check{S}_{l2}$ are two independently determined locally optimal solutions of P (see Subsection 2.2.3 p.13).

- $\check{S}_\varepsilon$ denotes a heuristic ε-penalty solution of P derived from the penalty method with respect to solution $\check{S}_{l1}$ (see Definition 2.2.4 p.13).

- $\{\check{S}_{1(\varepsilon)}, \check{S}_{2(\varepsilon)}\}$ denotes an ε-penalty solution-pair of P derived from the mutual penalty method (see Subsection 2.3.3.2 p.23).

Simulation

- For $t = 1, \ldots, N$, $P^{(t)} = (E, F, w^{(t)})$ is a randomly generated instance of P (i.i.d for all t) and $\widehat{w}^{(t)}$ is a randomly generated instance of perturbed weights of problem $P^{(t)}$ (i.i.d for all t).
- The solutions of problem instance $P^{(t)}$ are marked by a superscript (t).
- $I_\varepsilon := \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\} \subset \mathbb{R}$ with $0 < \varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_N$ denotes the set of penalty parameters ε for a certain experiment.

A.1.2 Definitions for Exact Algorithms

A.1.2.1 Penalty Method

$$\bar{\varphi}_\varepsilon := \frac{\frac{1}{T}\sum_{t=1\ldots T}\ \min\left(\widehat{w}^{(t)}(S_0^{(t)}), \widehat{w}^{(t)}(S_\varepsilon^{(t)})\right)}{\frac{1}{T}\sum_{t=1\ldots T}\ \widehat{w}^{(t)}(S_0^{(t)})} \tag{A.1}$$

$$\delta_\varepsilon : \quad P\{|\varphi_\varepsilon - \bar{\varphi}_\varepsilon| < \delta_\varepsilon\} \approx 0.98 \tag{A.2}$$

$$r_\varepsilon^< := \frac{1}{T}\sum_{t=1\ldots T} 1_{(\widehat{w}^{(t)}(S_\varepsilon^{(t)})<\widehat{w}^{(t)}(S_0^{(t)}))} \tag{A.3}$$

$$r_\varepsilon^= := \frac{1}{T}\sum_{t=1\ldots T} 1_{(\widehat{w}^{(t)}(S_\varepsilon^{(t)})=\widehat{w}^{(t)}(S_0^{(t)}))} \tag{A.4}$$

$$\varepsilon_* := \varepsilon_{i^*} \ : \ \bar{\varphi}_{\varepsilon_{i^*}} \leq \bar{\varphi}_{\varepsilon_i} \quad \text{for all} \quad \varepsilon_i, \varepsilon_{i^*} \in I_\varepsilon \tag{A.5}$$

$$\triangle\bar{\varphi}_{0.1} := \bar{\varphi}_{\varepsilon_*+0.1} - \bar{\varphi}_{\varepsilon_*} \tag{A.6}$$

A.1.2.2 Mutual Penalty Method

$$\bar{\varphi}_\varepsilon^m := \frac{\frac{1}{T}\sum_{t=1\ldots T}\ \min\left(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}), \widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)})\right)}{\frac{1}{T}\sum_{t=1\ldots T}\ \widehat{w}^{(t)}(S_0^{(t)})} \tag{A.7}$$

$$\delta_\varepsilon^m : \quad P\{|\varphi_\varepsilon^m - \bar{\varphi}_\varepsilon^m| < \delta_\varepsilon^m\} \approx 0.98 \tag{A.8}$$

$$r_\varepsilon^< := \frac{1}{T}\sum_{t=1\ldots T} 1_{(\min(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}),\widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)}))<\widehat{w}^{(t)}(S_0^{(t)}))} \tag{A.9}$$

$$r_\varepsilon^= := \frac{1}{T}\sum_{t=1\ldots T} 1_{(\min(\widehat{w}^{(t)}(S_{1(\varepsilon)}^{(t)}),\widehat{w}^{(t)}(S_{2(\varepsilon)}^{(t)}))=\widehat{w}^{(t)}(S_0^{(t)}))} \tag{A.10}$$

$$\varepsilon_*^m := \varepsilon_{i^*} \ : \ \bar{\varphi}_{\varepsilon_{i^*}}^m \leq \bar{\varphi}_{\varepsilon_i}^m \quad \text{for all} \quad \varepsilon_i, \varepsilon_{i^*} \in I_\varepsilon \tag{A.11}$$

$$\bar{\varphi}_{\varepsilon_*}^m := \bar{\varphi}_{\varepsilon_*^m}^m \tag{A.12}$$

$$\delta_{\varepsilon_*}^m := \delta_{\varepsilon_*^m}^m \tag{A.13}$$

$$\triangle\bar{\varphi}_{0.1}^m := \bar{\varphi}_{\varepsilon_*^m+0.1}^m - \bar{\varphi}_{\varepsilon_*^m}^m \tag{A.14}$$

A.1.3 Definitions for Heuristic Algorithms

A.1.3.1 Penalty Method

$$\bar{\chi}_{\varepsilon} := \frac{\frac{1}{T}\sum_{t=1\dots T}\ \min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}), \widehat{w}^{(t)}(\check{S}^{(t)}_{\varepsilon}))}{\frac{1}{T}\sum_{t=1\dots T}\ \min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}), \widehat{w}^{(t)}(\check{S}^{(t)}_{l2}))} \tag{A.15}$$

$$\check{\delta}_{\varepsilon}: \quad P\{|\chi_{\varepsilon} - \bar{\chi}_{\varepsilon}| < \check{\delta}_{\varepsilon}\} \approx 0.98 \tag{A.16}$$

$$r^{<}_{\varepsilon} := \frac{1}{T}\sum_{t=1\dots T} 1_{(\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{\varepsilon}))<\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{l2})))} \tag{A.17}$$

$$r^{=}_{\varepsilon} := \frac{1}{T}\sum_{t=1\dots T} 1_{(\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{\varepsilon}))=\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{l2})))} \tag{A.18}$$

$$\varepsilon_* := \varepsilon_{i^*} \,:\, \bar{\chi}_{\varepsilon_{i^*}} \le \bar{\chi}_{\varepsilon_i} \quad \text{for all} \quad \varepsilon_i, \varepsilon_{i^*} \in I_{\varepsilon} \tag{A.19}$$

$$\triangle\bar{\chi}_{0.1} := \bar{\chi}_{\varepsilon_*+0.1} - \bar{\chi}_{\varepsilon_*} \tag{A.20}$$

A.1.3.2 Mutual Penalty Method

$$\bar{\chi}^{m}_{\varepsilon} := \frac{\frac{1}{T}\sum_{t=1\dots T}\ \min(\widehat{w}^{(t)}(\check{S}^{(t)}_{1(\varepsilon)}), \widehat{w}^{(t)}(\check{S}^{(t)}_{2(\varepsilon)}))}{\frac{1}{T}\sum_{t=1\dots T}\ \min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}), \widehat{w}^{(t)}(\check{S}^{(t)}_{l2}))} \tag{A.21}$$

$$\check{\delta}^{m}_{\varepsilon}: \quad P\{|\chi^{m}_{\varepsilon} - \bar{\chi}^{m}_{\varepsilon}| < \check{\delta}^{m}_{\varepsilon}\} \approx 0.98 \tag{A.22}$$

$$r^{<}_{\varepsilon} := \frac{1}{T}\sum_{t=1\dots T} 1_{(\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{1(\varepsilon)}),\widehat{w}^{(t)}(\check{S}^{(t)}_{2(\varepsilon)}))<\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{l2})))} \tag{A.23}$$

$$r^{=}_{\varepsilon} := \frac{1}{T}\sum_{t=1\dots T} 1_{(\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{1(\varepsilon)}),\widehat{w}^{(t)}(\check{S}^{(t)}_{2(\varepsilon)}))=\min(\widehat{w}^{(t)}(\check{S}^{(t)}_{l1}),\widehat{w}^{(t)}(\check{S}^{(t)}_{l2})))} \tag{A.24}$$

$$\varepsilon^{m}_* := \varepsilon_{i^*} \,:\, \bar{\chi}^{m}_{\varepsilon_{i^*}} \le \bar{\chi}^{m}_{\varepsilon_i} \quad \text{for all} \quad \varepsilon_i, \varepsilon_{i^*} \in I_{\varepsilon} \tag{A.25}$$

$$\bar{\chi}^{m}_{\varepsilon_*} := \bar{\chi}^{m}_{\varepsilon^{m}_*} \tag{A.26}$$

$$\check{\delta}^{m}_{\varepsilon_*} := \check{\delta}^{m}_{\varepsilon^{m}_*} \tag{A.27}$$

$$\triangle\bar{\chi}^{m}_{0.1} := \bar{\chi}^{m}_{\varepsilon^{m}_*+0.1} - \bar{\chi}^{m}_{\varepsilon^{m}_*} \tag{A.28}$$

A.2 Local Search

Local Search is a well known heuristic algorithm (see [AL 2003]). There are many different variants of Local Search. Here we briefly explain the variant that we used for our experiments.

Consider a sum type optimization problem $P = (E, F, w)$. For every solution $S \subseteq F$ let $N(S)$ be the set of neighbors of S and let $\mathcal{N} = \{N(S) \ : \ S \in F\}$. Algorithm A.1 describes a basic local search. If F is finite, the algorithm stops after a finite number of steps.

Require: $P = (E, F, w)$, $\mathcal{N}$, initial solution $S^{(0)} \in F$
$t := 0$;
calculate $w(S^{(t)})$;
while (not done) **do**
  search $S^{(t+1)} \in N(S^{(t)})$ with $w(S^{(t+1)}) - w(S^{(t)}) < 0$
  if found **then**
    $t := t + 1$;
  else
    done;
solution $S^{(t)}$ is locally optimal for P

Algorithm A.1: basic local search

The algorithm does not specify the search procedure exactly. In a real application it has to be specified how the initial solution $S^{(0)}$ is chosen and how the local improvement steps are done. In our experiments the initial solution $S^{(0)}$ is generated randomly. For the local improvement steps we use the following strategy:

- Evaluate the neighbors $N(S)$ of a solution S in a certain order. The first neighbor that results in an improvement is chosen. This strategy is called *fast local search*. Here a further differentiation of the algorithm is possible by the order in which the neighbors are evaluated.

  In our experiments we use a cyclical strategy with a random starting position. Let the set of neighbors of S be $N(S) = \{S_0, S_1, \ldots, S_{k-1}\}$, the S_i are numbered. Furthermore, let i be a random number that is uniformly distributed in $\{0, 1, \ldots, k-1\}$. Then we evaluate the neighbors in the order $S_i, S_{(i+1) \bmod k}, \ldots, S_{(i+k-1) \bmod k}$. The random starting position i is generated independently for every search step.

Finally, for a specific optimization problem the structure of the neighborhood $\mathcal{N} = \{N(S) \ : \ S \in F\}$ has to be defined. For the assignment problem we use the 2-exchange neighborhood (see [AL 2003] p.4) and for the traveling salesman problem we use the 2-opt neighborhood (see [AL 2003] p.230).

Appendix B

Results – $(p; 1, c)$-Model

B.1 Exact Algorithms

B.1.1 Penalty Method (see Sections 3.3.1, 3.5, 4.1)

B.1.1.1 Shortest Path Problem

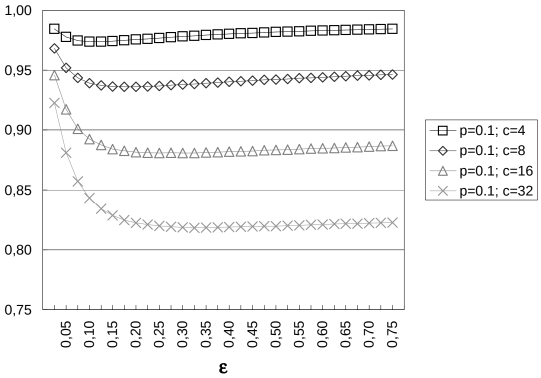


Figure B.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^5$]

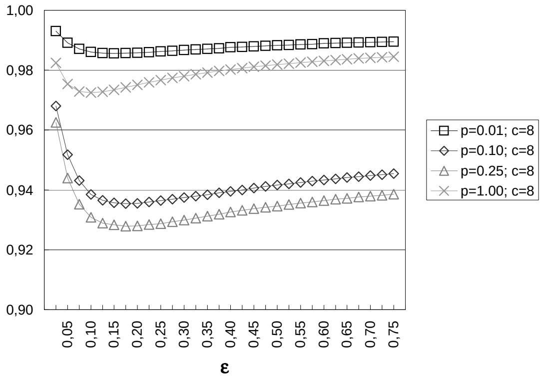


Figure B.2: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^5$]

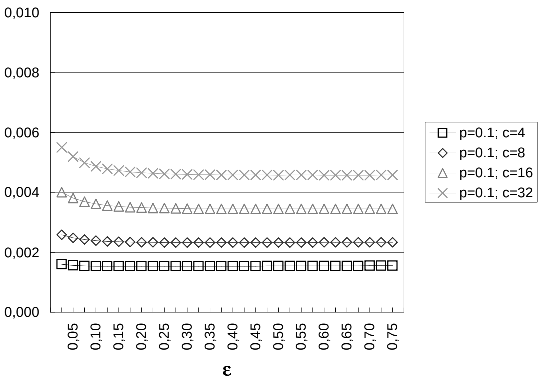


Figure B.3: Shortest path problem – 98% certain maximal error δ_ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^5$]

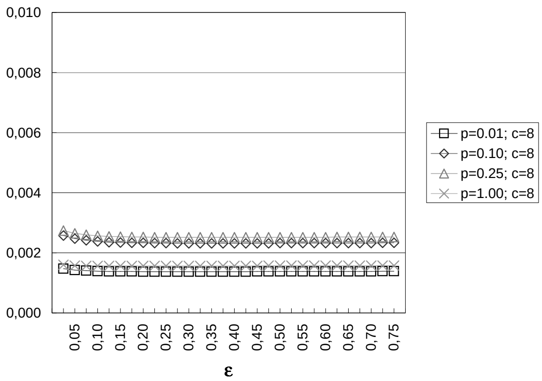


Figure B.4: Shortest path problem – 98% certain maximal error δ_ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^5$]

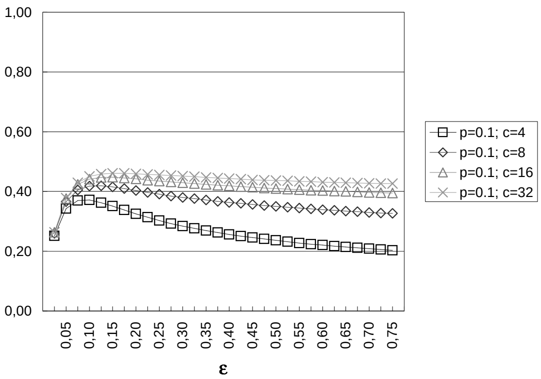


Figure B.5: Shortest path problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$;
$[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

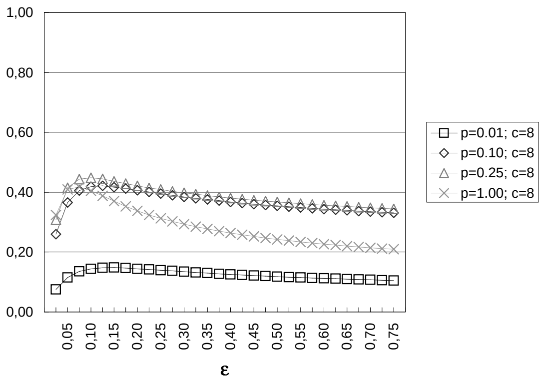


Figure B.6: Shortest path problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$;
$[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

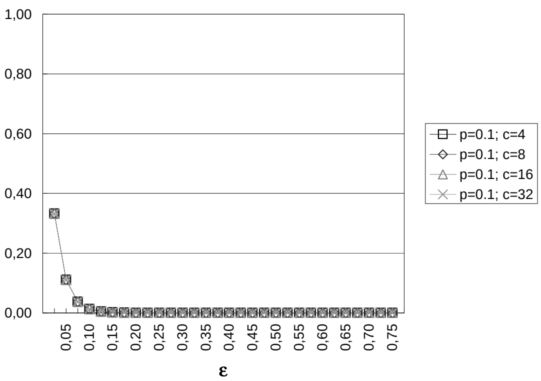


Figure B.7: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$;
$[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

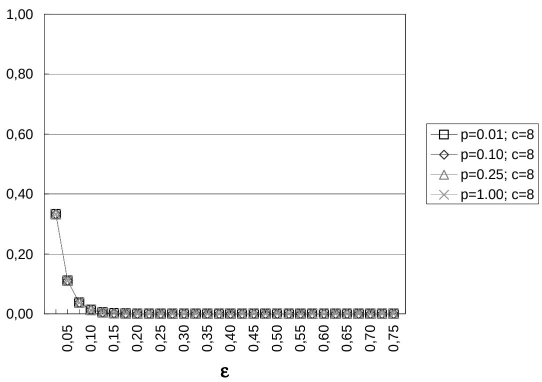


Figure B.8: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$;
$[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

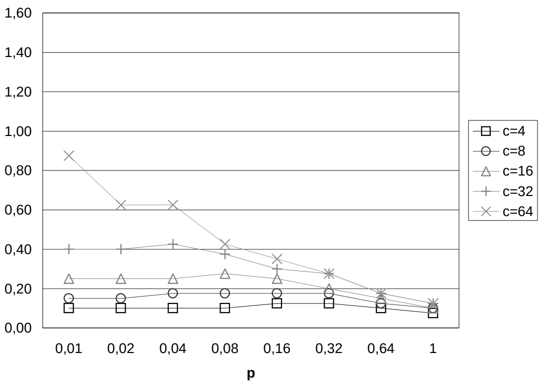


Figure B.9: Shortest path problem – optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

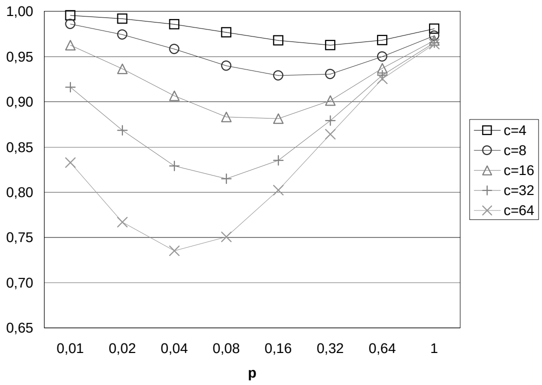


Figure B.10: Shortest path problem – optimal averages time rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

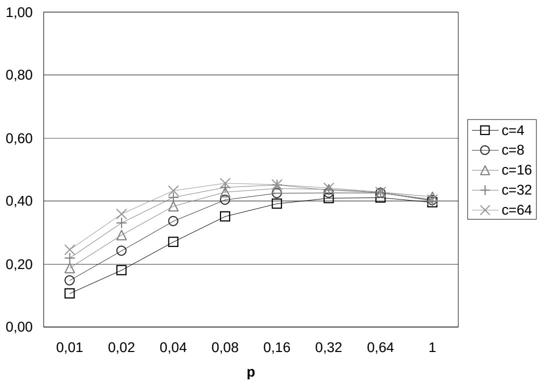


Figure B.11: Shortest path problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(p, c)$ with $\widehat{w}(S_{\varepsilon_*}) < \widehat{w}(S_0)$ $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

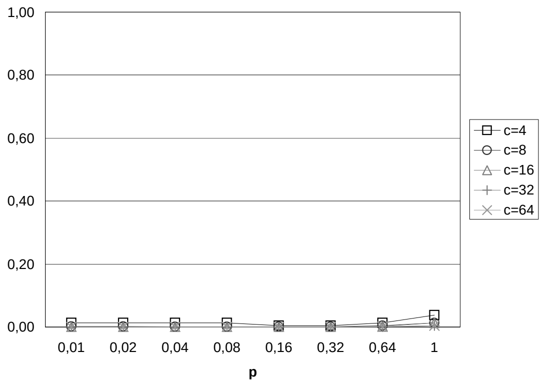


Figure B.12: Shortest path problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(p, c)$ with $\widehat{w}(S_{\varepsilon_*}) = \widehat{w}(S_0)$ $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

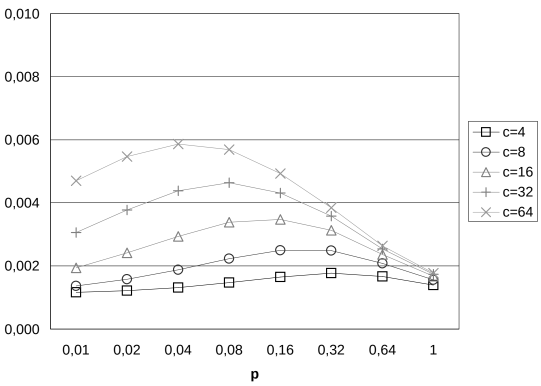


Figure B.13: Shortest path problem – 98% certain maximal error $\delta_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

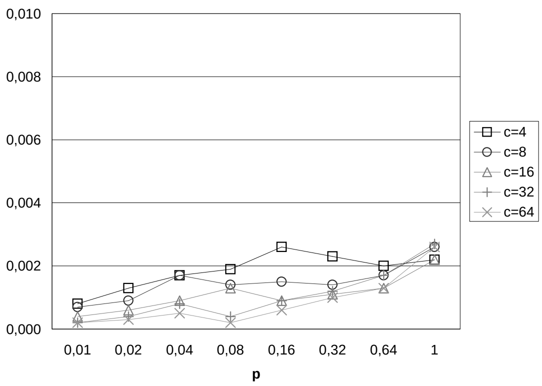


Figure B.14: Shortest path problem – loss of quality $\triangle\bar{\varphi}_{0.1}(p, c)$ by overestimating ε_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

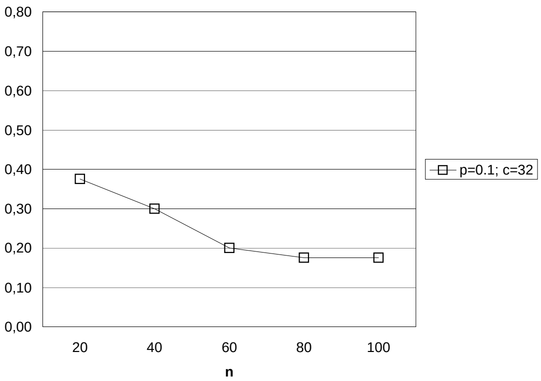


Figure B.15: Shortest path problem – optimal penalty parameters $\varepsilon_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

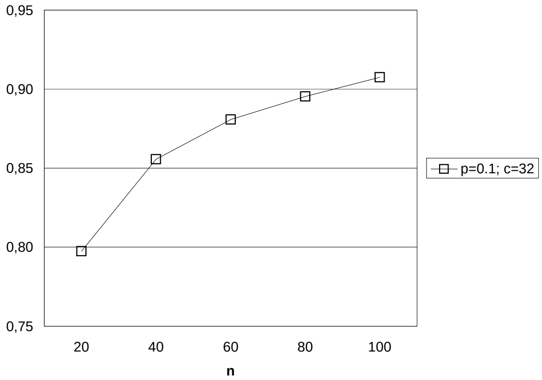


Figure B.16: Shortest path problem – optimal average time rates $\bar{\varphi}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

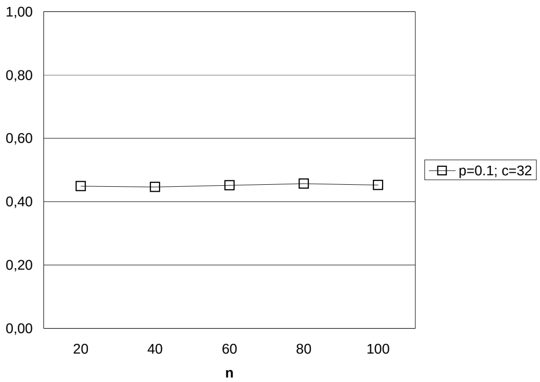


Figure B.17: Shortest path problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(n)$ with $\widehat{w}(S_{\varepsilon_*}) < \widehat{w}(S_0)$ $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

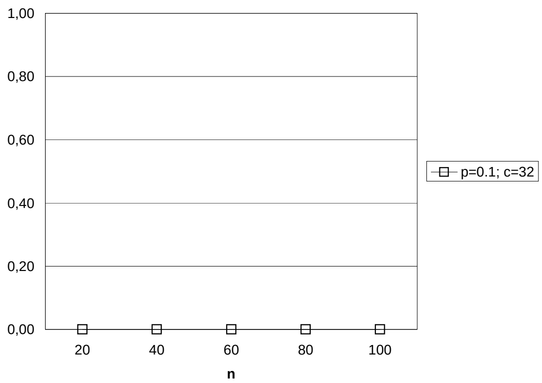


Figure B.18: Shortest path problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(n)$ with $\widehat{w}(S_{\varepsilon_*}) = \widehat{w}(S_0)$ $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

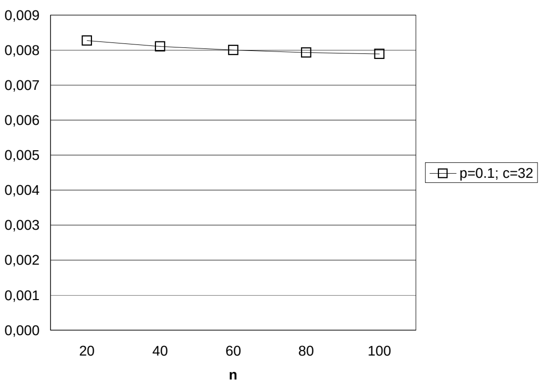


Figure B.19: Shortest path problem – 98% certain maximal error $\delta_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

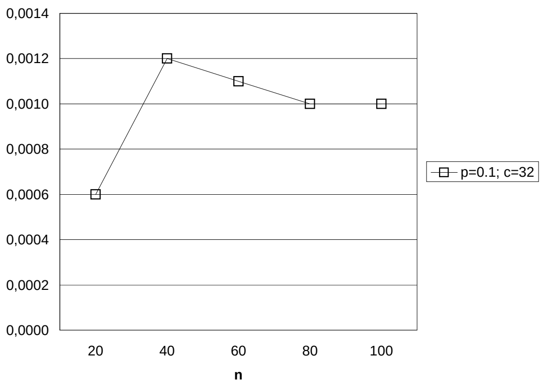


Figure B.20: Shortest path problem – loss of quality $\triangle\bar{\varphi}_{0.1}(n)$ by overestimating ε_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.00\};\ T = 10^5]$

B.1.1.2 Assignment Problem

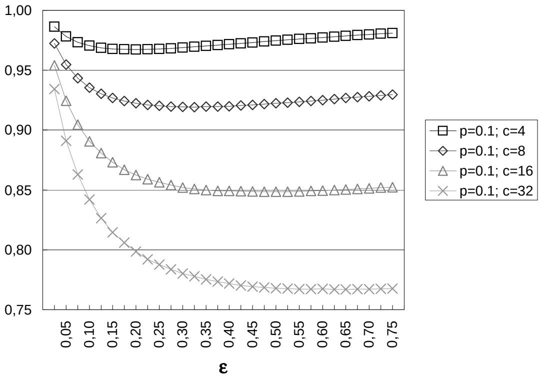


Figure B.21: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

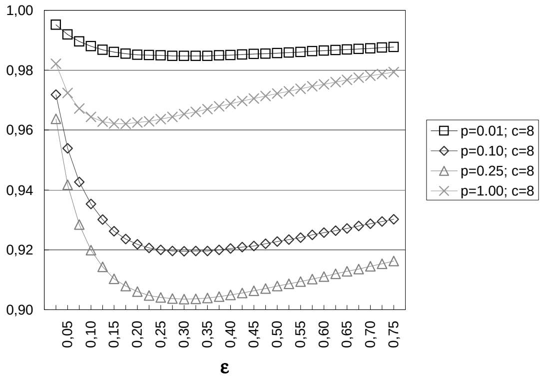


Figure B.22: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

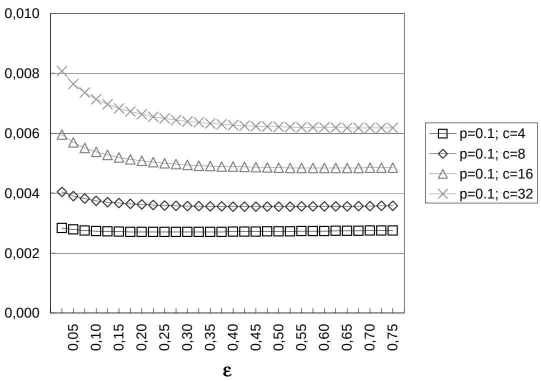


Figure B.23: Assignment problem – 98% certain maximal error δ_ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

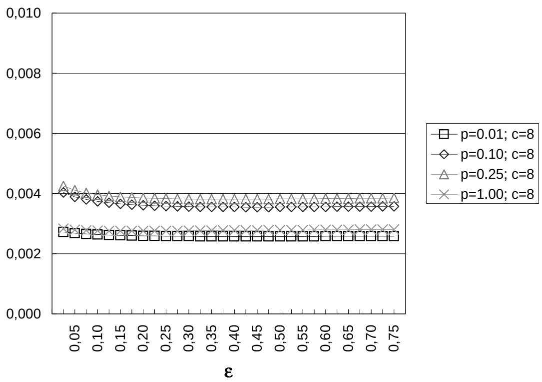


Figure B.24: Assignment problem – 98% certain maximal error δ_ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

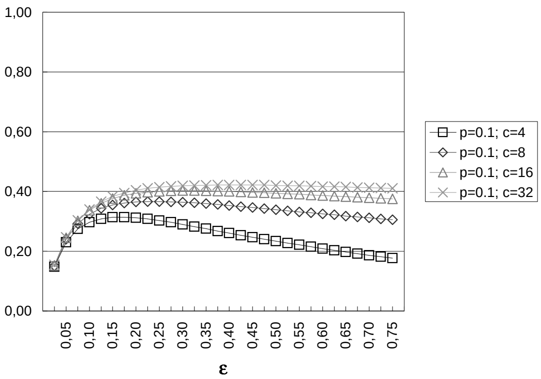


Figure B.25: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

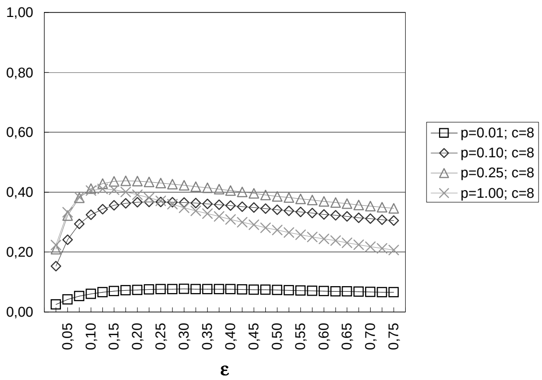


Figure B.26: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\widehat{w}(S_\varepsilon) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

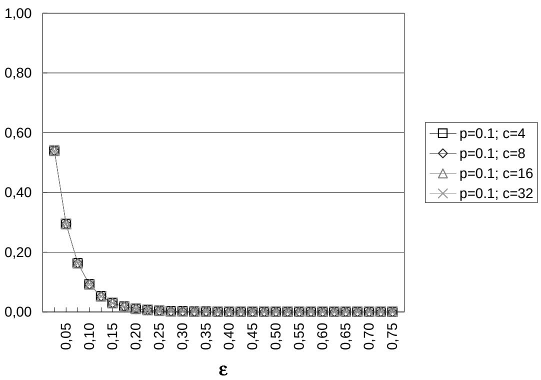


Figure B.27: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

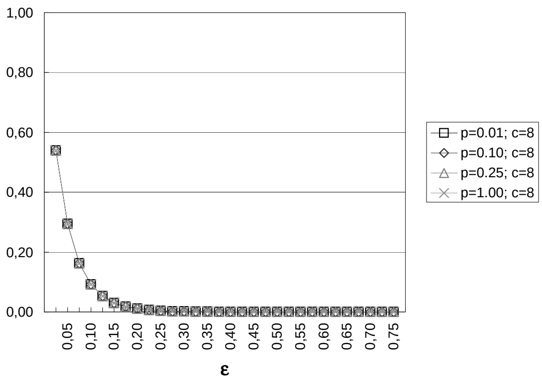


Figure B.28: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\widehat{w}(S_\varepsilon) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

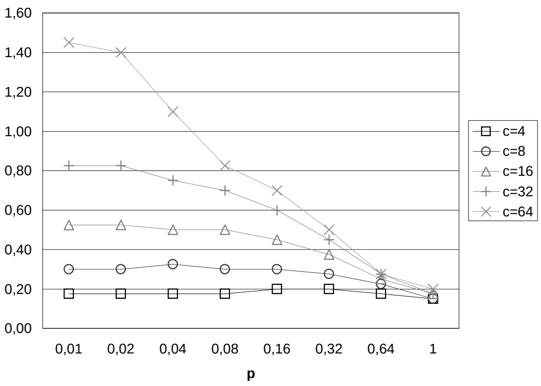


Figure B.29: Assignment problem – optimal penalty parameters $\varepsilon_*(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

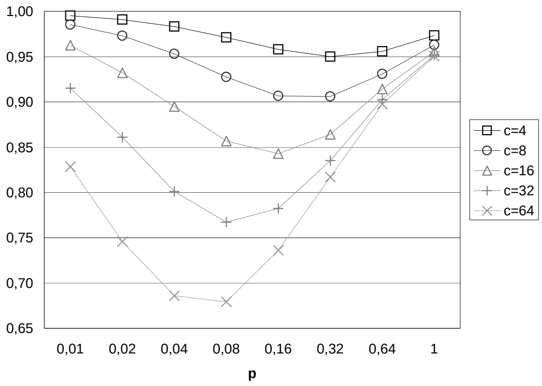


Figure B.30: Assignment problem – optimal average cost rates $\bar{\varphi}_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

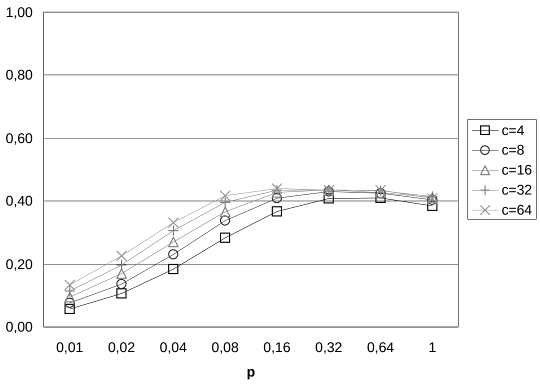


Figure B.31: Assignment problem – relative part of simulation runs $r^<_{\varepsilon_*}(p,c)$ with $\widehat{w}(S_{\varepsilon_*}) < \widehat{w}(S_0)$ $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

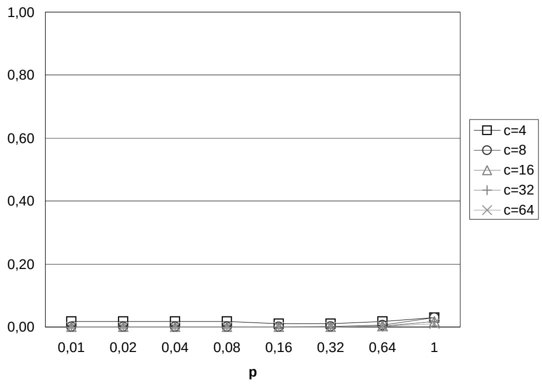


Figure B.32: Assignment problem – relative part of simulation runs $r^=_{\varepsilon_*}(p,c)$ with $\widehat{w}(S_{\varepsilon_*}) = \widehat{w}(S_0)$ $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

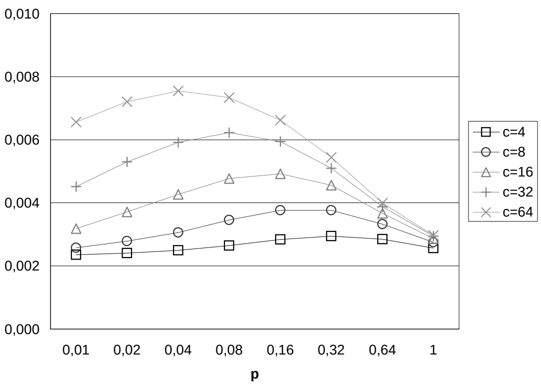


Figure B.33: Assignment problem – 98% certain maximal error $\delta_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

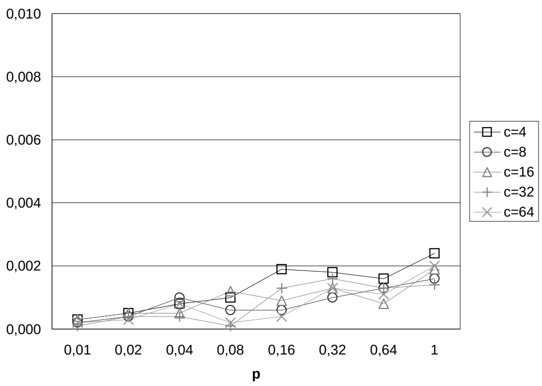


Figure B.34: Assignment problem – loss of quality $\triangle\bar{\varphi}_{0.1}(p,c)$ by overestimating ε_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

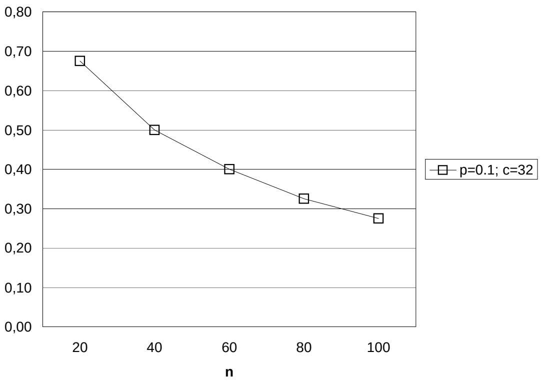


Figure B.35: Assignment problem – optimal penalty parameters $\varepsilon_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

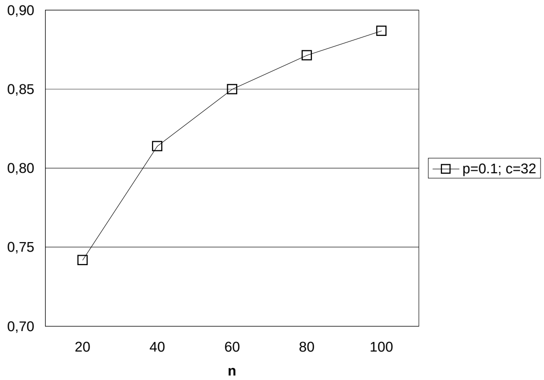


Figure B.36: Assignment problem – optimal average cost rates $\bar{\varphi}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

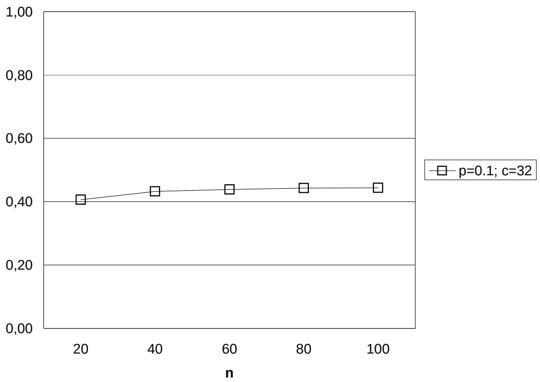


Figure B.37: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(n)$ with $\widehat{w}(S_{\varepsilon_*}) < \widehat{w}(S_0)$ $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

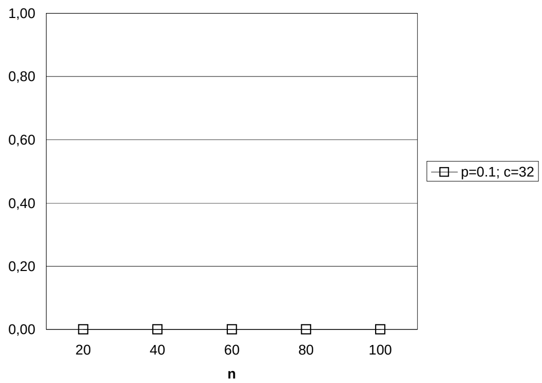


Figure B.38: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(n)$ with $\widehat{w}(S_{\varepsilon_*}) = \widehat{w}(S_0)$ $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

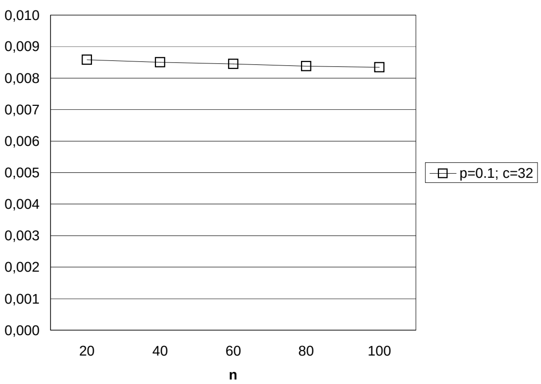


Figure B.39: Assignment problem – 98% certain maximal error $\delta_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

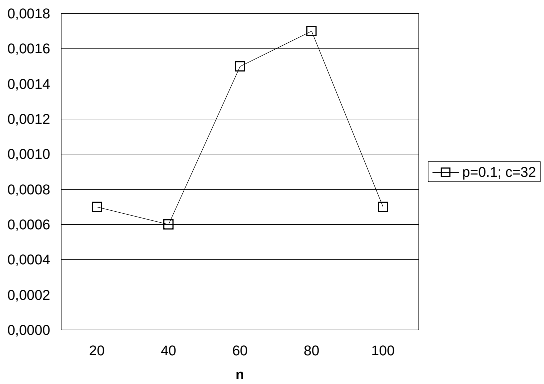


Figure B.40: Assignment problem – loss of quality $\triangle\bar{\varphi}_{0.1}(n)$ by overestimating ε_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^5]$

B.1.2 Mutual Penalty Method (see Sections 3.3.2, 3.5, 4.1)

B.1.2.1 Shortest Path Problem

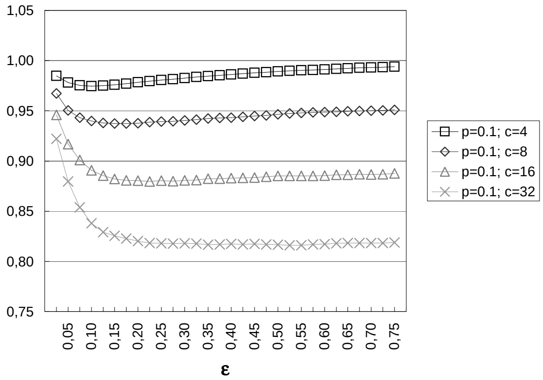


Figure B.41: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

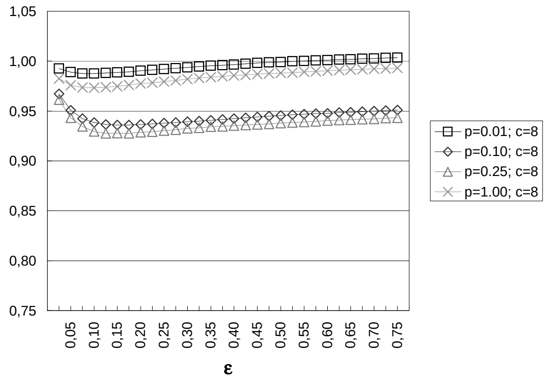


Figure B.42: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

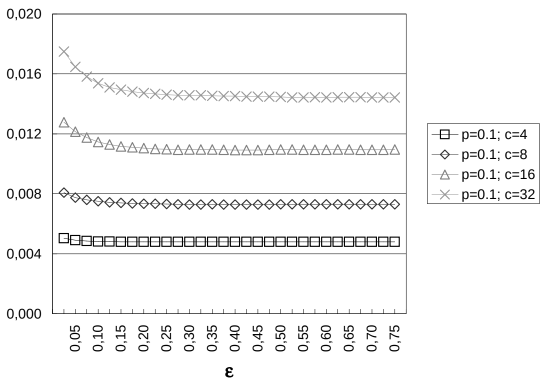


Figure B.43: Shortest path problem – 98% certain maximal error δ_{ε}^{m}; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

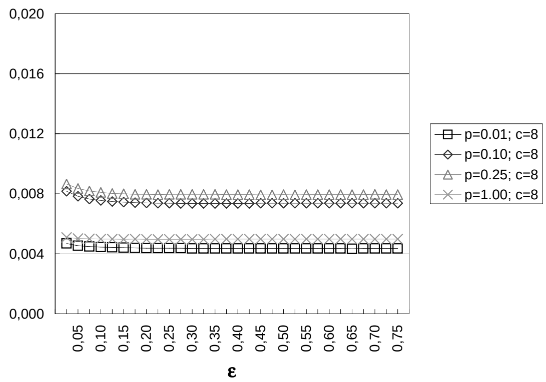


Figure B.44: Shortest path problem – 98% certain maximal error δ_{ε}^{m}; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

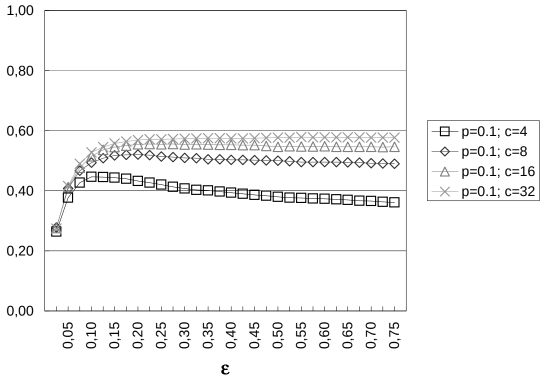


Figure B.45: Shortest path problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

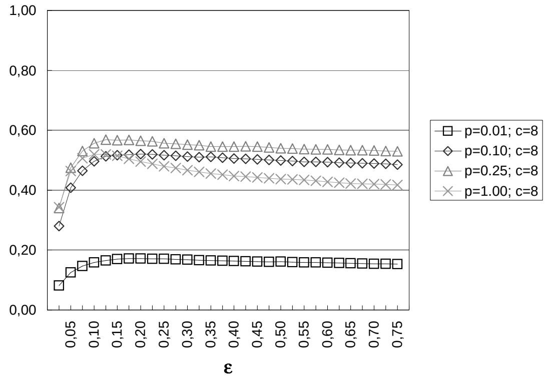


Figure B.46: Shortest path problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

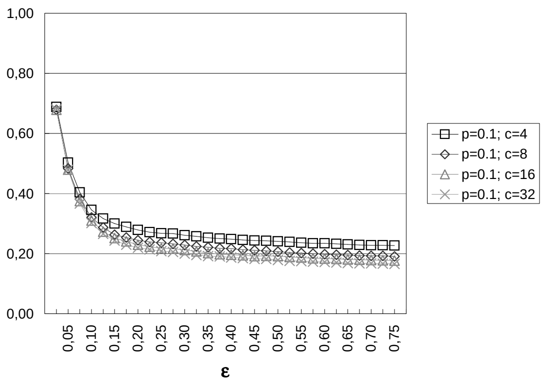


Figure B.47: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

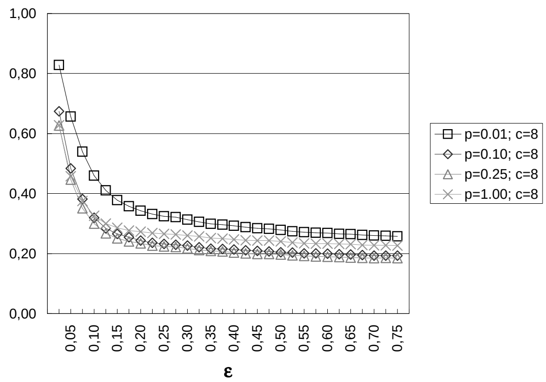


Figure B.48: Shortest path problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

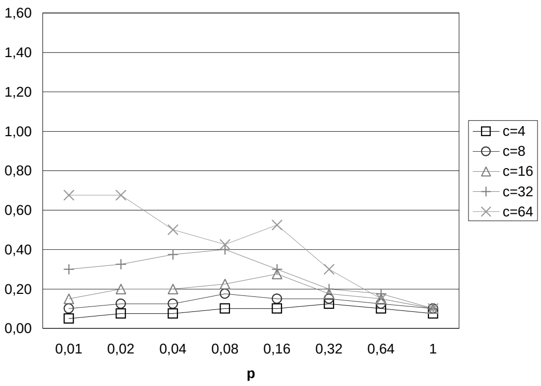


Figure B.49: Shortest path problem – optimal penalty parameters $\varepsilon^m_*(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

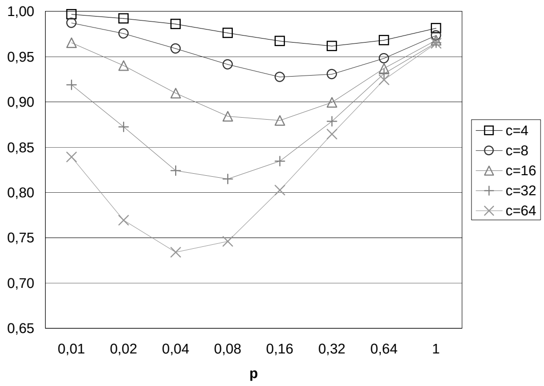


Figure B.50: Shortest path problem – optimal average time rates $\bar{\varphi}^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

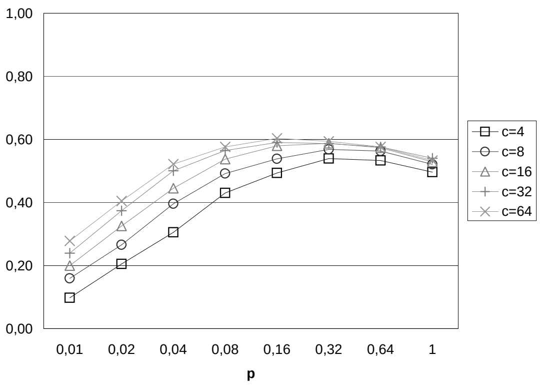


Figure B.51: Shortest path problem – relative part of simulation runs $r^{<}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) < \widehat{w}(S_0)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

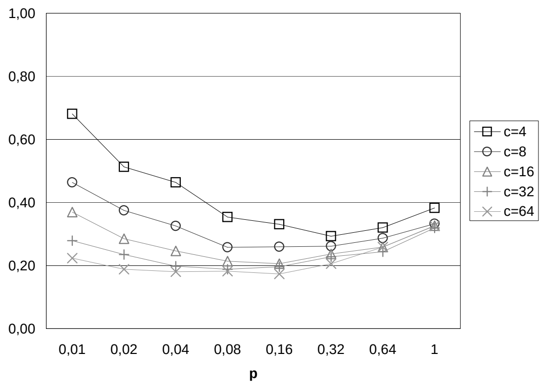


Figure B.52: Shortest path problem – relative part of simulation runs $r^{=}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) = \widehat{w}(S_0)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

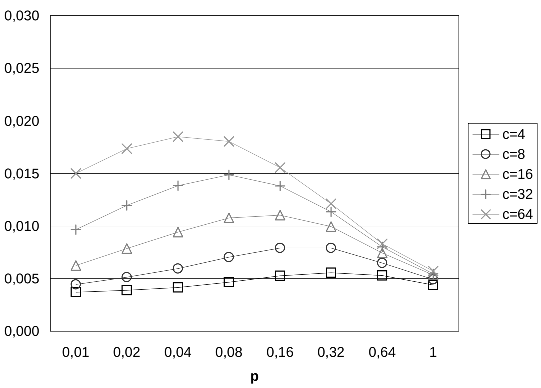


Figure B.53: Shortest path problem – 98% certain maximal error $\delta^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

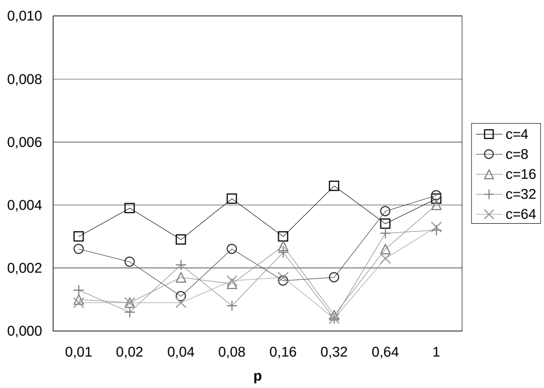


Figure B.54: Shortest path problem – loss of quality $\triangle\bar{\varphi}^m_{0.1}(p,c)$ by overestimating ε^m_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

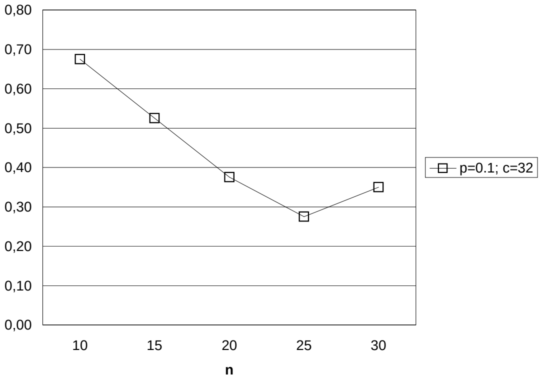


Figure B.55: Shortest path problem – optimal penalty parameters $\varepsilon_*^m(n)$; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

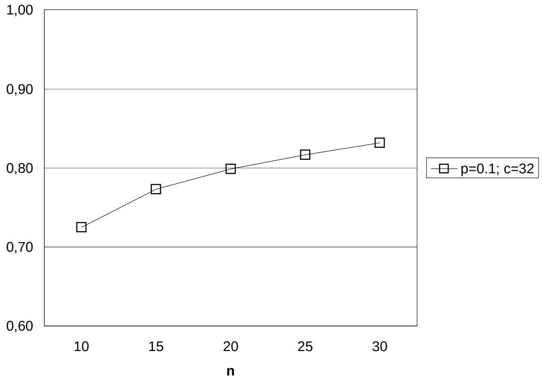


Figure B.56: Shortest path problem – optimal average time rates $\bar{\varphi}_{\varepsilon_*}^m(n)$; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

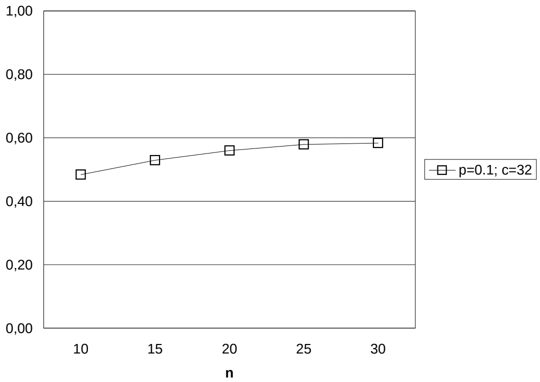


Figure B.57: Shortest path problem – relative part of simulation runs $r_{\varepsilon_*^m}^{<}(n)$ with $\min(\widehat{w}(S_{1(\varepsilon_*^m)}), \widehat{w}(S_{2(\varepsilon_*^m)})) < \widehat{w}(S_0)$; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

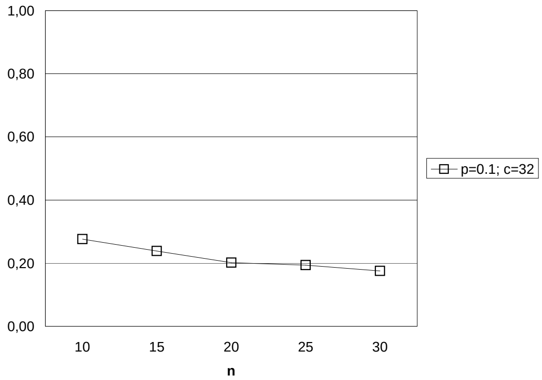


Figure B.58: Shortest path problem – relative part of simulation runs $r_{\varepsilon_*^m}^{=}(n)$ with $\min(\widehat{w}(S_{1(\varepsilon_*^m)}), \widehat{w}(S_{2(\varepsilon_*^m)})) = \widehat{w}(S_0)$; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

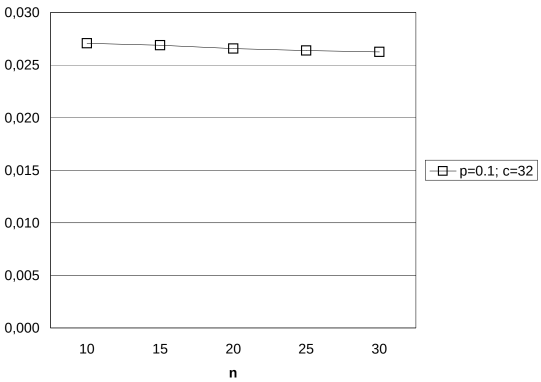


Figure B.59: Shortest path problem – 98% certain maximal error $\delta_{\varepsilon_*}^m(n)$; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

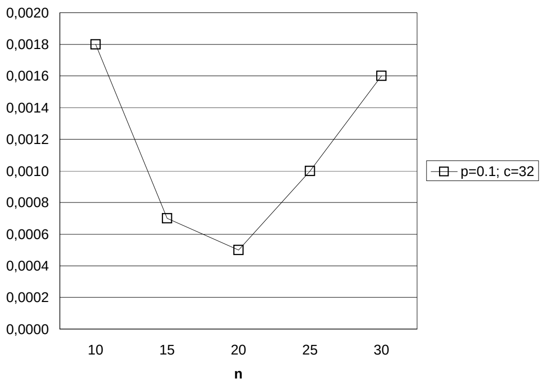


Figure B.60: Shortest path problem – loss of quality $\triangle\bar{\varphi}_{0.1}^m(n)$ by overestimating ε_*^m by 0.1; $[I_\varepsilon = \{0.025, 0.05, \ldots, 1.50\};\ T = 10^4]$

B.1.2.2 Assignment Problem

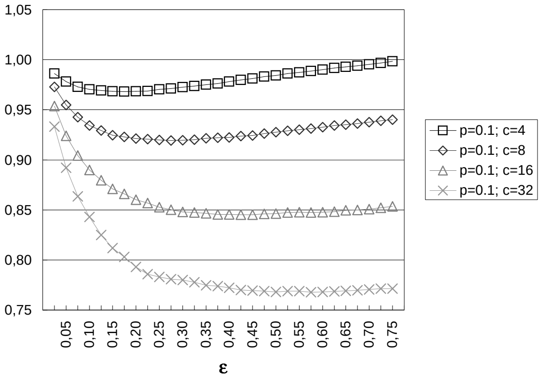


Figure B.61: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon^m$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

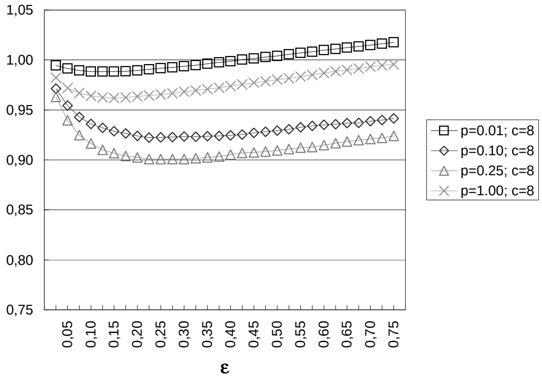


Figure B.62: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon^m$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

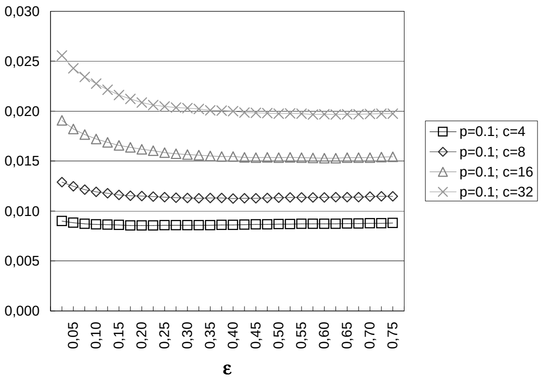


Figure B.63: Assignment problem – 98% certain maximal error δ_ε^m; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

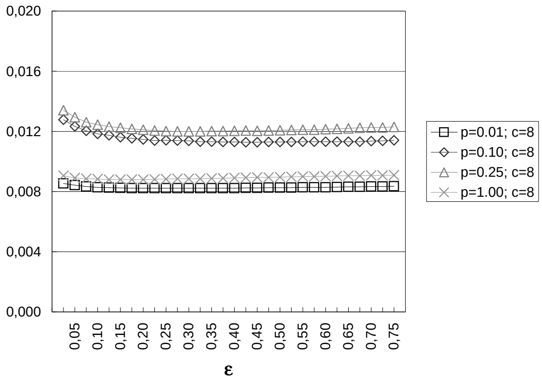


Figure B.64: Assignment problem – 98% certain maximal error δ_ε^m; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

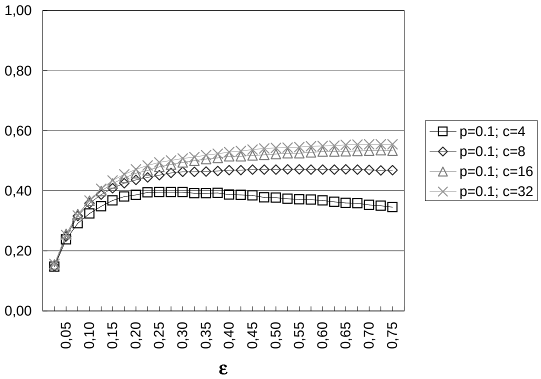


Figure B.65: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

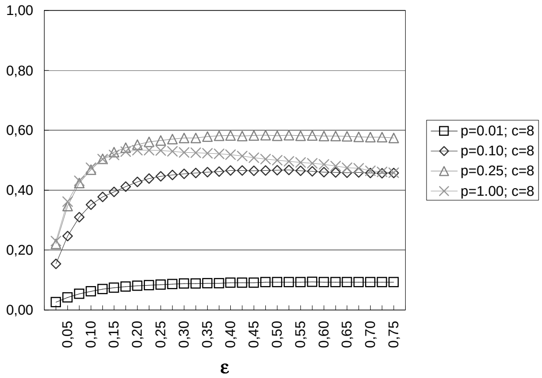


Figure B.66: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) < \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

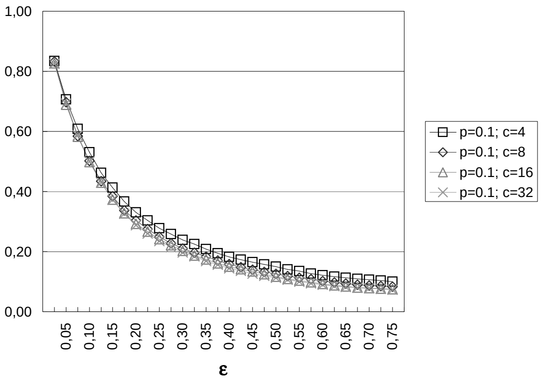


Figure B.67: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

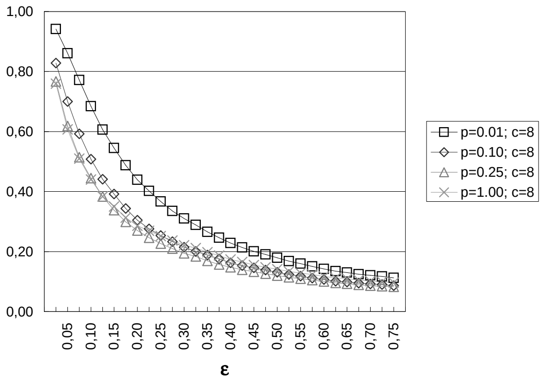


Figure B.68: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(S_{1(\varepsilon)}), \widehat{w}(S_{2(\varepsilon)})) = \widehat{w}(S_0)$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

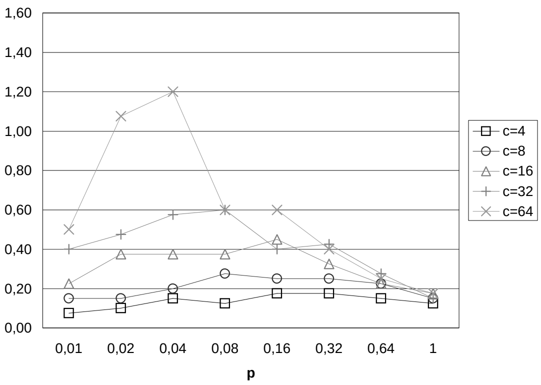


Figure B.69: Assignment problem – optimal penalty parameters $\varepsilon^m_*(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

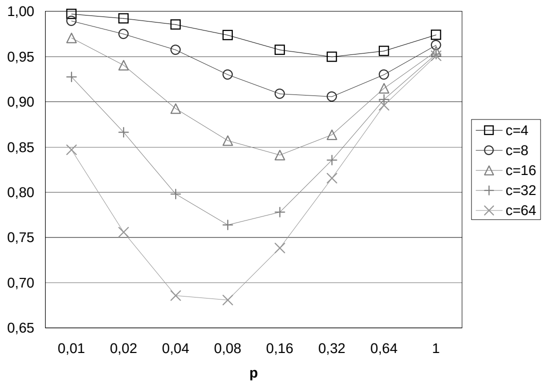


Figure B.70: Assignment problem – optimal average cost rates $\bar{\varphi}^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

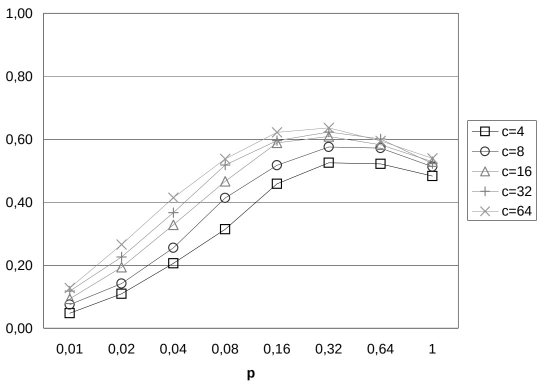


Figure B.71: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) < \widehat{w}(S_0)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

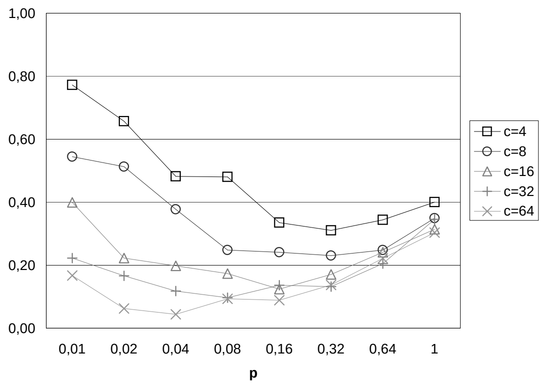


Figure B.72: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) = \widehat{w}(S_0)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

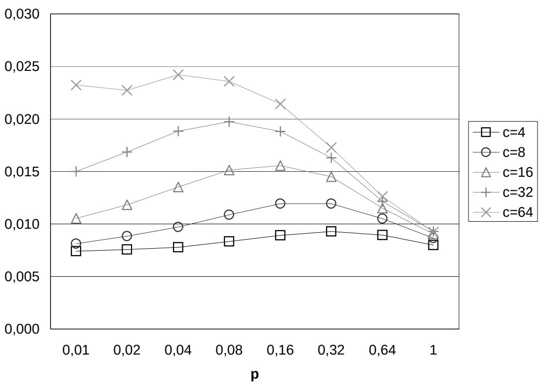


Figure B.73: Assignment problem – 98% certain maximal error $\delta^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

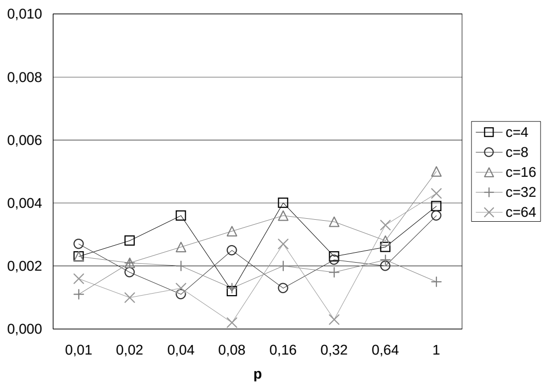


Figure B.74: Assignment problem – loss of quality $\triangle\bar{\varphi}^m_{0.1}(p,c)$ by overestimating ε^m_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

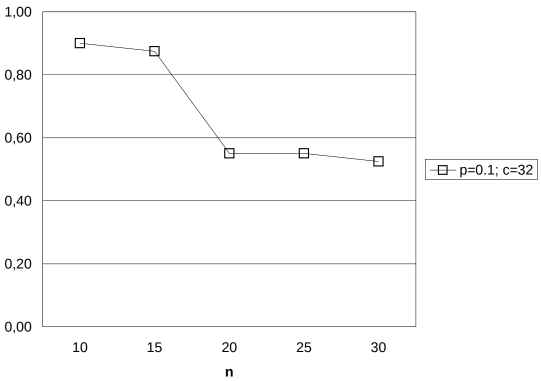


Figure B.75: Assignment problem – optimal penalty parameters $\varepsilon^m_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

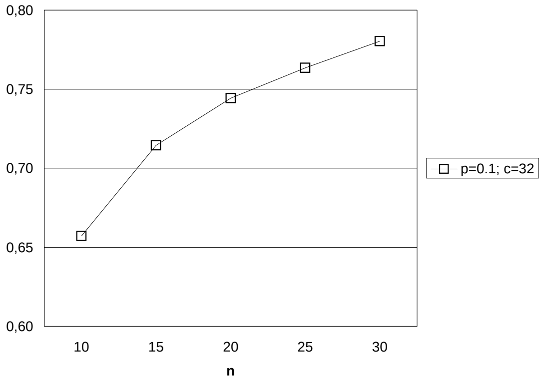


Figure B.76: Assignment problem – optimal average cost rates $\bar{\varphi}^m_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

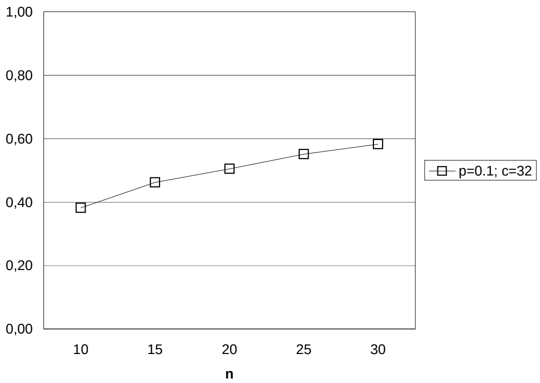


Figure B.77: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon^m_*}(n)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) < \widehat{w}(S_0)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

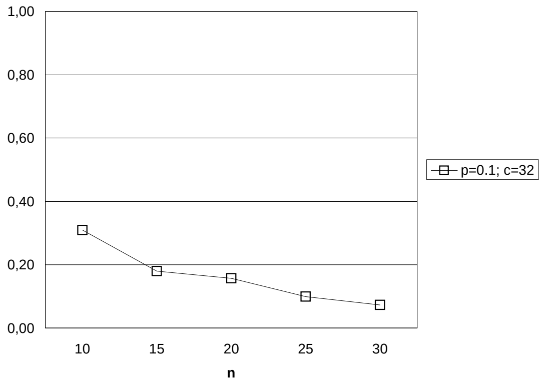


Figure B.78: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon^m_*}(n)$ with $\min(\widehat{w}(S_{1(\varepsilon^m_*)}), \widehat{w}(S_{2(\varepsilon^m_*)})) = \widehat{w}(S_0)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

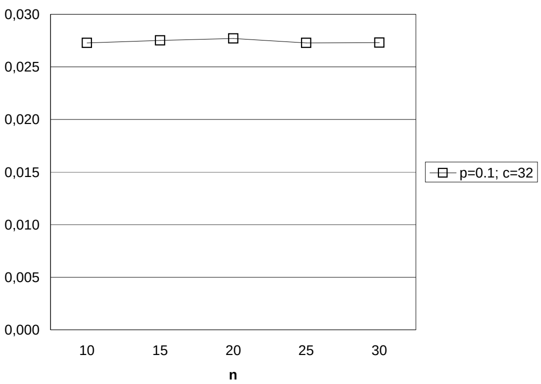


Figure B.79: Assignment problem – 98% certain maximal error $\delta^m_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

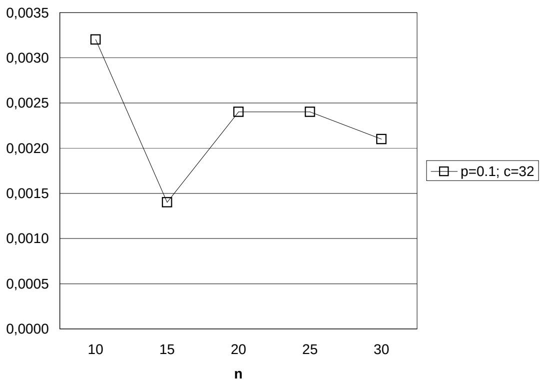


Figure B.80: Assignment problem – loss of quality $\triangle\bar{\varphi}^m_{0.1}(n)$ by overestimating ε^m_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.5\};\ T = 10^4]$

B.2 Heuristic Algorithms

B.2.1 Penalty Method (see Sections 3.4.1, 3.5, 4.1)

B.2.1.1 Assignment Problem

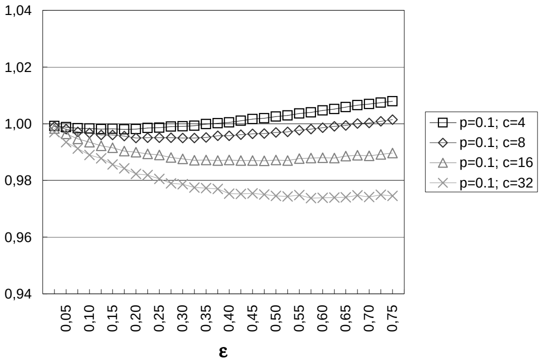


Figure B.81: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

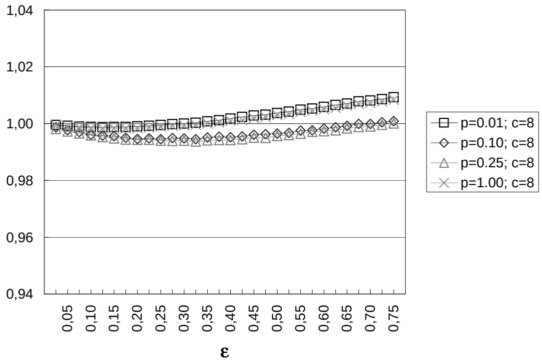


Figure B.82: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

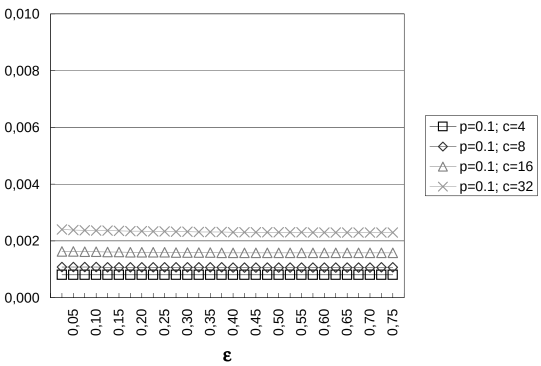


Figure B.83: Assignment problem – 98% certain maximal error $\check{\delta}_\varepsilon$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

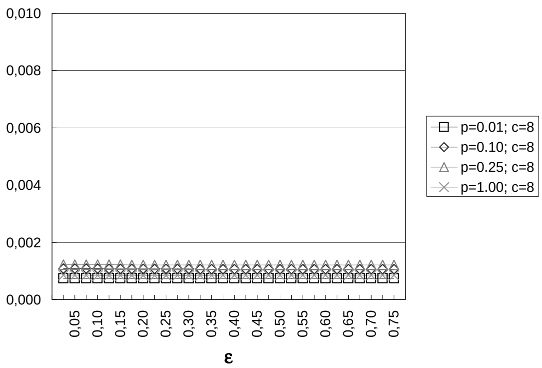


Figure B.84: Assignment problem – 98% certain maximal error $\check{\delta}_\varepsilon$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

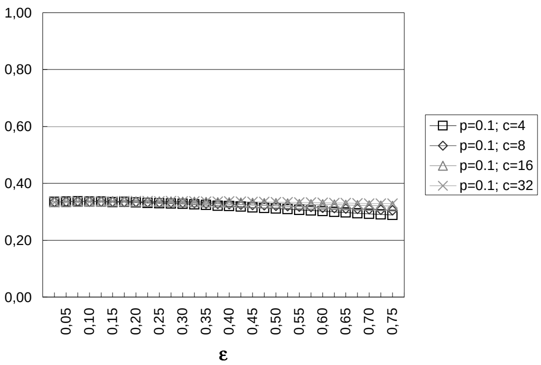


Figure B.85: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

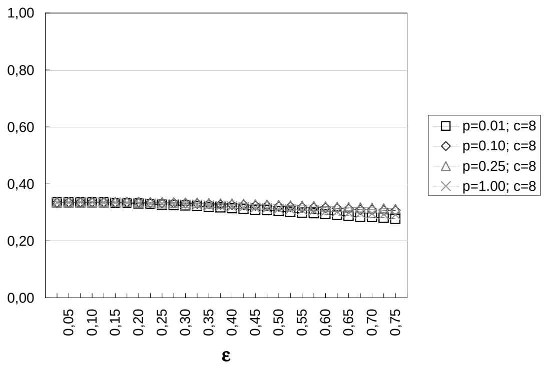


Figure B.86: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

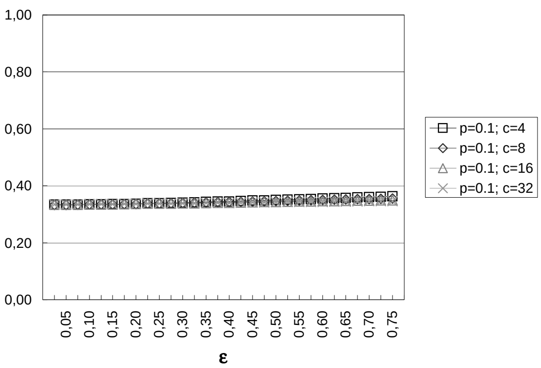


Figure B.87: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

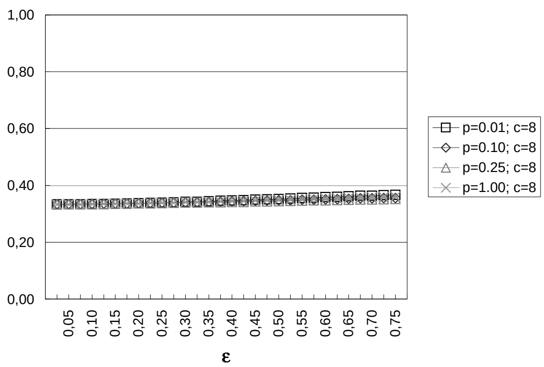


Figure B.88: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

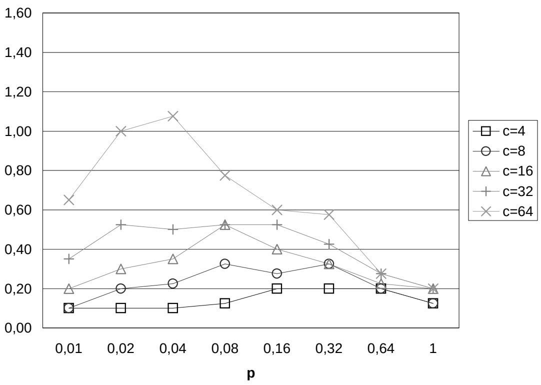


Figure B.89: Assignment problem – optimal penalty parameters $\varepsilon_*(p,c)$; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

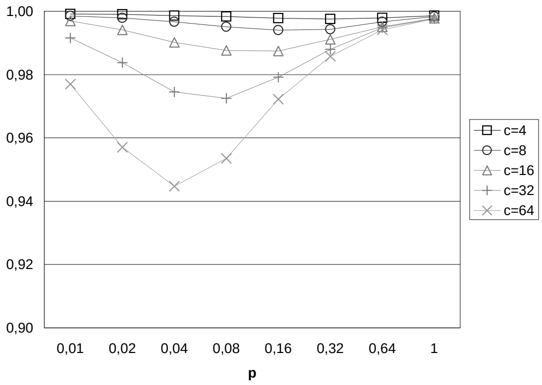


Figure B.90: Assignment problem – optimal average cost rates $\bar{\chi}_{\varepsilon_*}(p,c)$; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

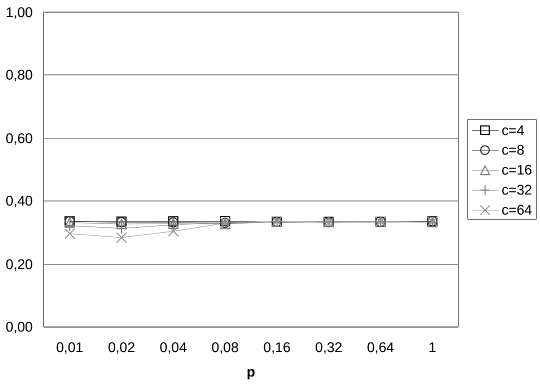


Figure B.91: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{l1}),\widehat{w}(\check{S}_{\varepsilon_*})) < \min(\widehat{w}(\check{S}_{l1}),\widehat{w}(\check{S}_{l2}))$; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

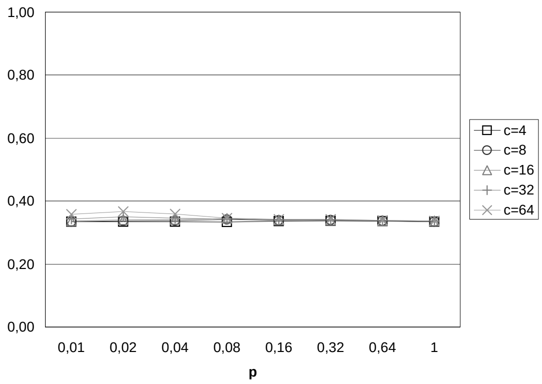


Figure B.92: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{l1}),\widehat{w}(\check{S}_{\varepsilon_*})) = \min(\widehat{w}(\check{S}_{l1}),\widehat{w}(\check{S}_{l2}))$; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

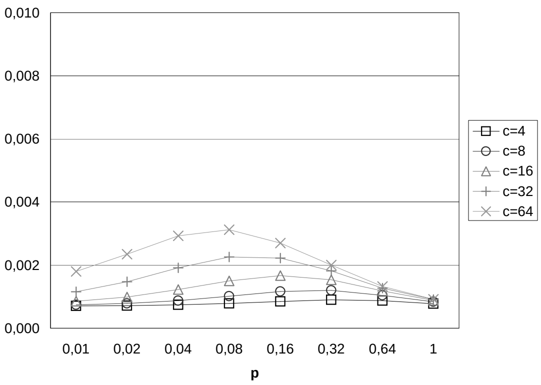


Figure B.93: Assignment problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}(p,c)$; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

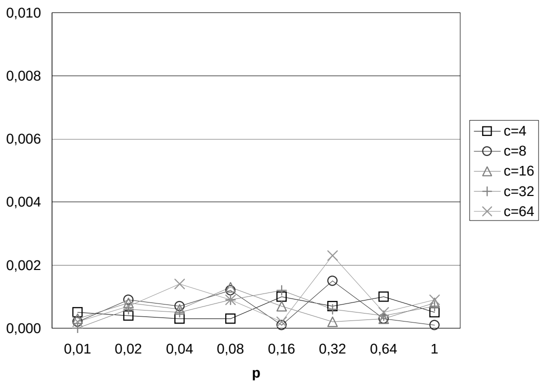


Figure B.94: Assignment problem – loss of quality $\triangle\bar{\chi}_{0.1}(p,c)$ by overestimating ε_* by 0.1; $[n=25;\ I_\varepsilon=\{0.025,0.05,\ldots,2.50\};\ T=10^6]$

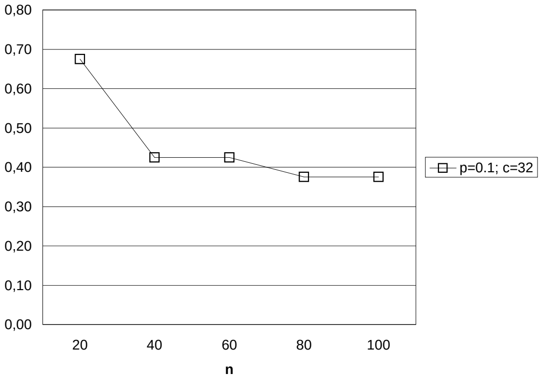


Figure B.95: Assignment problem – optimal penalty parameters $\varepsilon_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

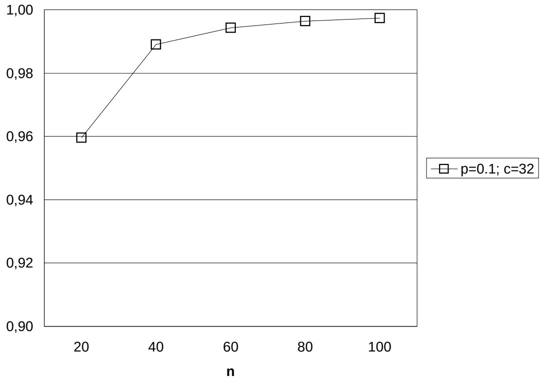


Figure B.96: Assignment problem – optimal average cost rates $\bar{\chi}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

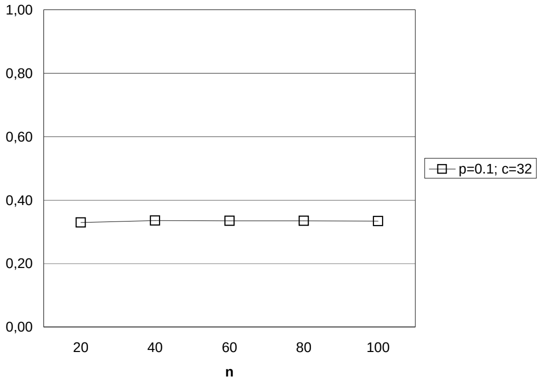


Figure B.97: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(n)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

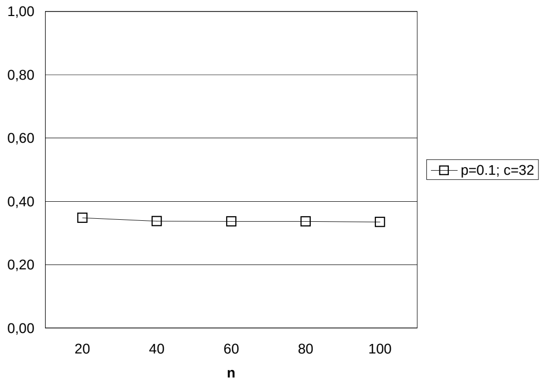


Figure B.98: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(n)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

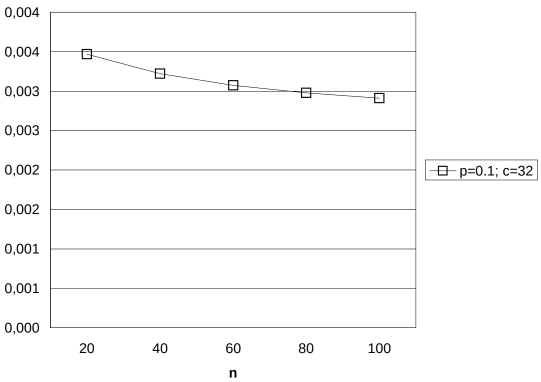


Figure B.99: Assignment problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

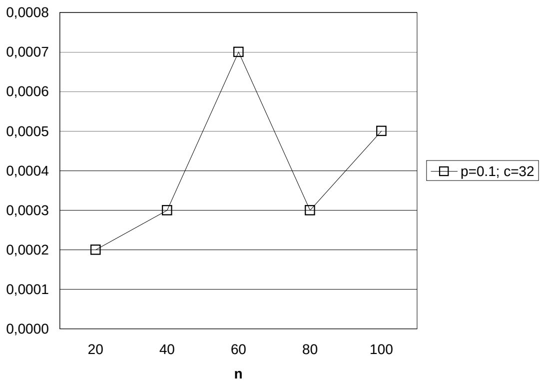


Figure B.100: Assignment problem – loss of quality $\triangle\bar{\chi}_{0.1}(n)$ by overestimating ε_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\}; \, T = 10^6]$

B.2.1.2 Traveling Salesman Problem

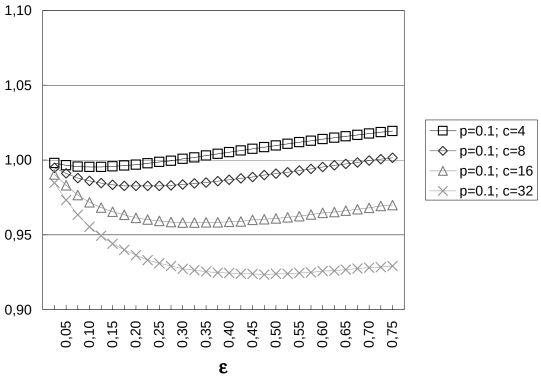


Figure B.101: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

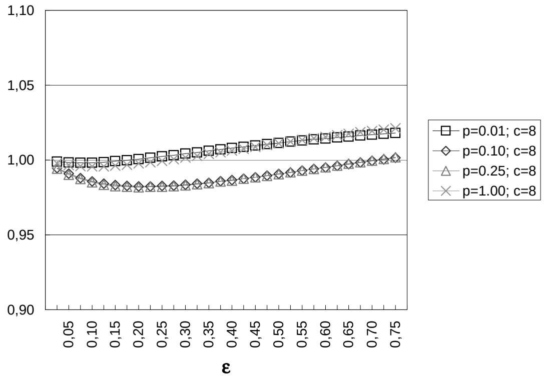


Figure B.102: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

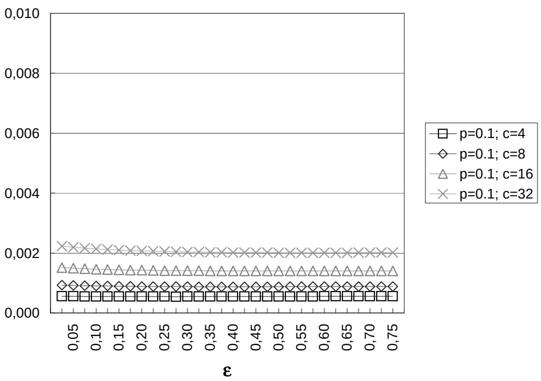


Figure B.103: Traveling salesman problem – 98% certain maximal error $\check{\delta}_\varepsilon$; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

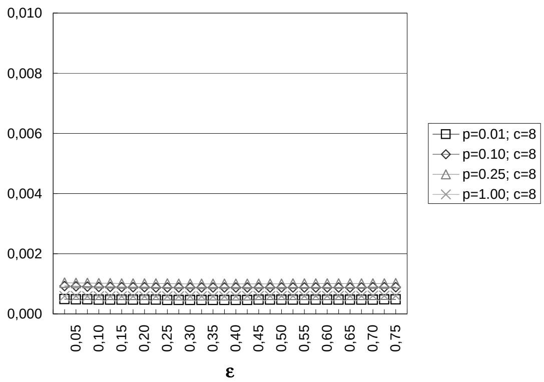


Figure B.104: Traveling salesman problem – 98% certain maximal error $\check{\delta}_\varepsilon$; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

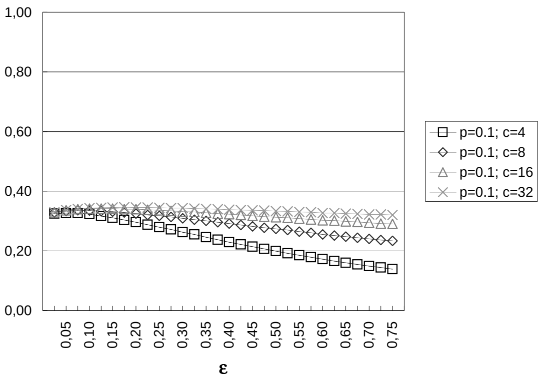


Figure B.105: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

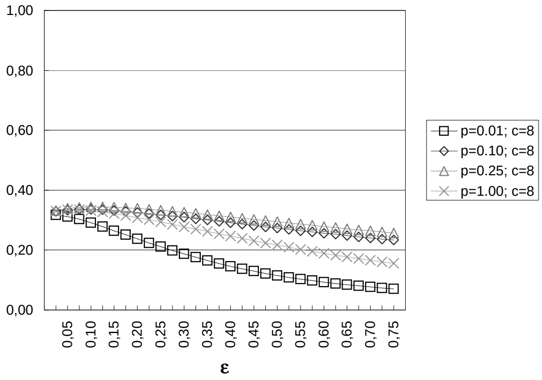


Figure B.106: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

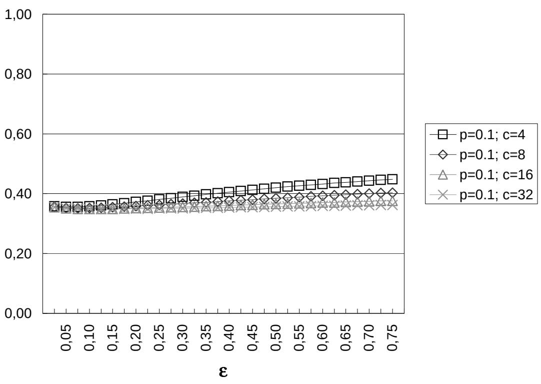


Figure B.107: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

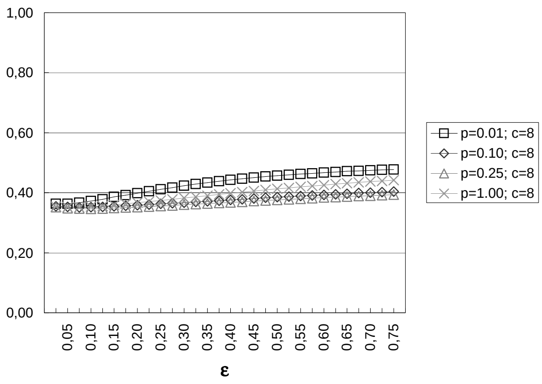


Figure B.108: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_\varepsilon)) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

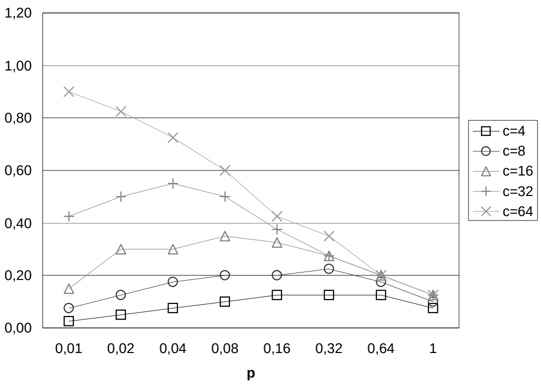


Figure B.109: Traveling salesman problem – optimal penalty parameters $\varepsilon_*(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

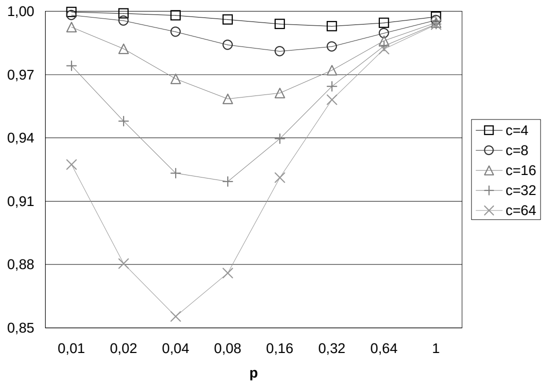


Figure B.110: Traveling salesman problem – optimal average time rates $\bar{\chi}_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

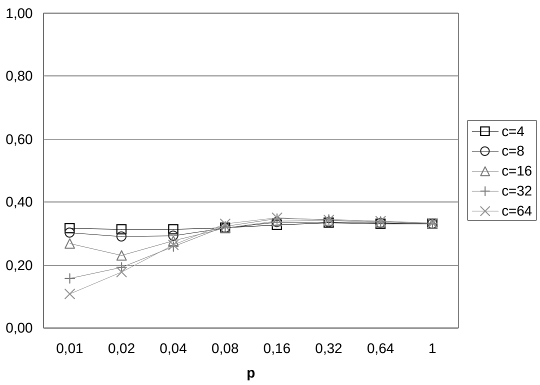


Figure B.111: Traveling salesman problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

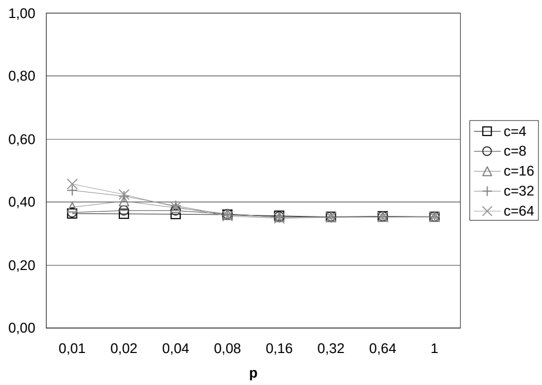


Figure B.112: Traveling salesman problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

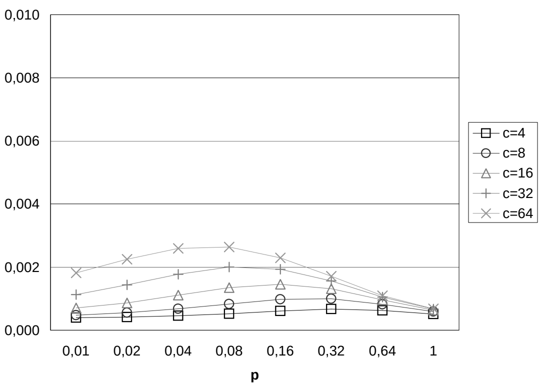


Figure B.113: Traveling salesman problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

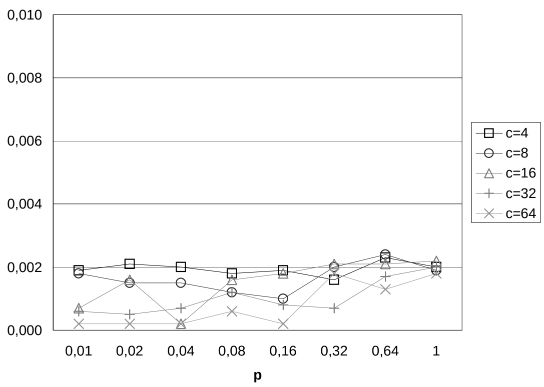


Figure B.114: Traveling salesman problem – loss of quality $\triangle\bar{\chi}_{0.1}(p,c)$ by overestimating ε_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \dots, 2.50\};\ T = 10^6]$

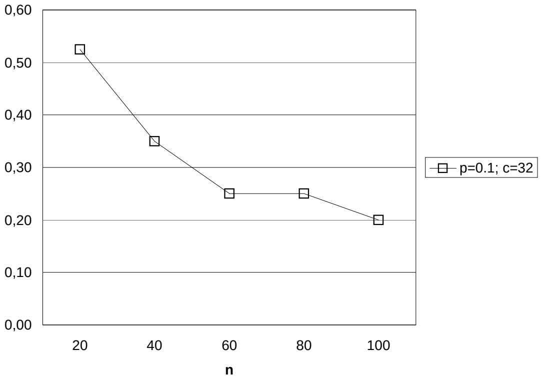


Figure B.115: Traveling salesman problem – optimal penalty parameters $\varepsilon_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

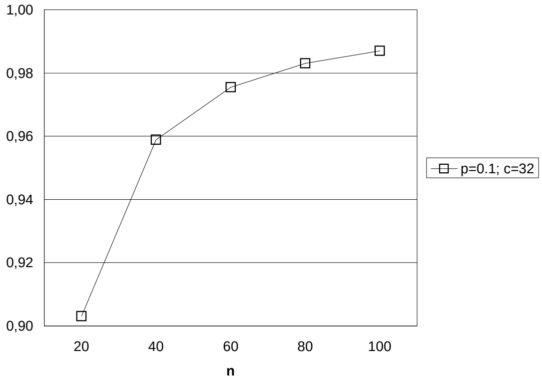


Figure B.116: Traveling salesman problem – optimal average time rates $\bar{\chi}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

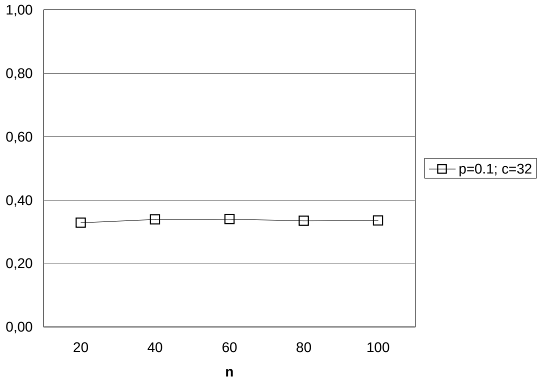


Figure B.117: Traveling salesman problem – relative part of simulation runs $r^{<}_{\varepsilon_*}(n)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

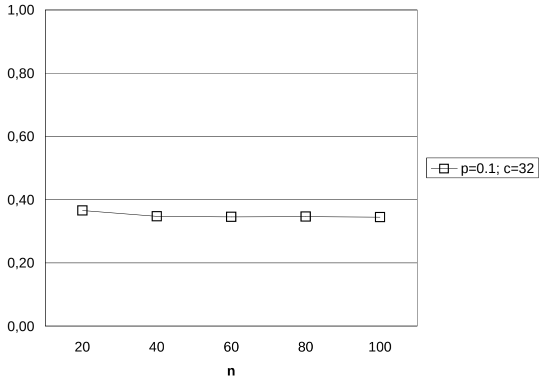


Figure B.118: Traveling salesman problem – relative part of simulation runs $r^{=}_{\varepsilon_*}(n)$ with $\min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{\varepsilon_*})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

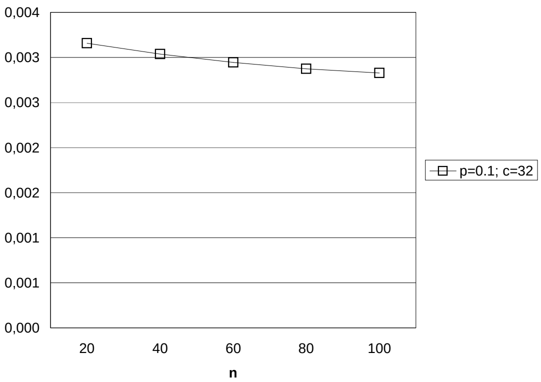


Figure B.119: Traveling salesman problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

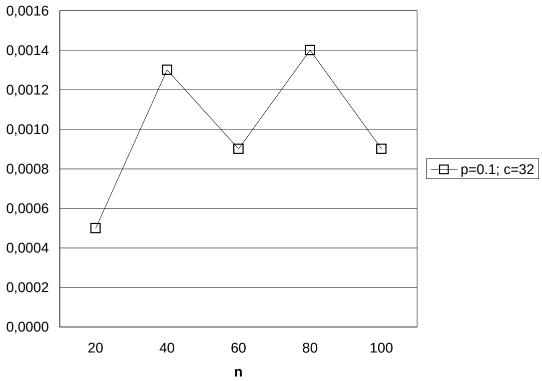


Figure B.120: Traveling salesman problem – loss of quality $\triangle\bar{\chi}_{0.1}(n)$ by overestimating ε_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

B.2.2 Mutual Penalty Method (see Sections 3.4.2, 3.5, 4.1)

B.2.2.1 Assignment Problem

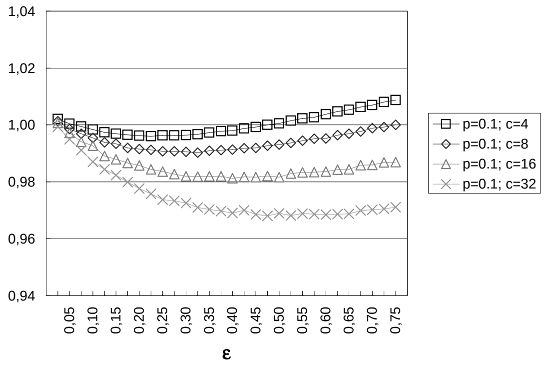


Figure B.121: Assignment problem – average cost rates $\bar{\chi}^m_\varepsilon$ for different ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

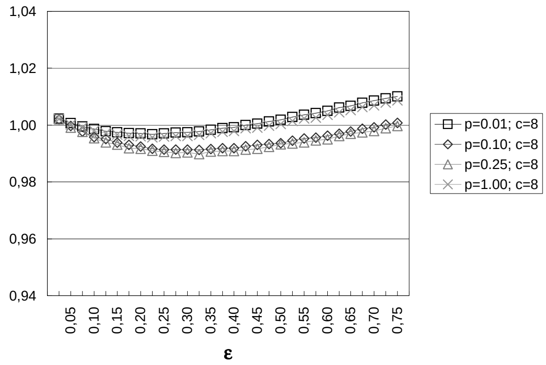


Figure B.122: Assignment problem – average cost rates $\bar{\chi}^m_\varepsilon$ for different ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

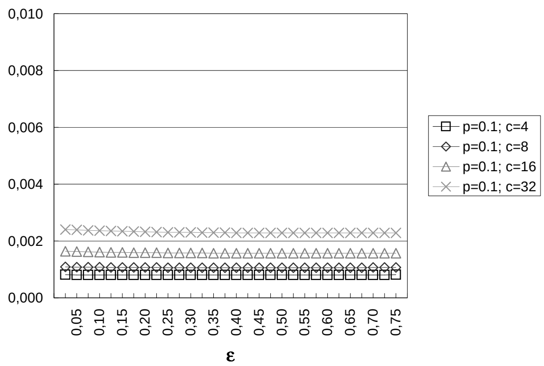


Figure B.123: Assignment problem – 98% certain maximal error $\check{\delta}^m_\varepsilon$; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

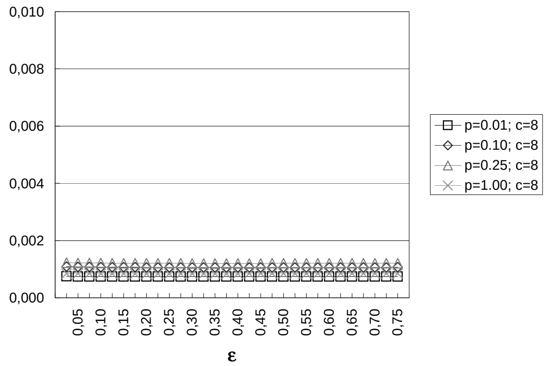


Figure B.124: Assignment problem – 98% certain maximal error $\check{\delta}^m_\varepsilon$; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

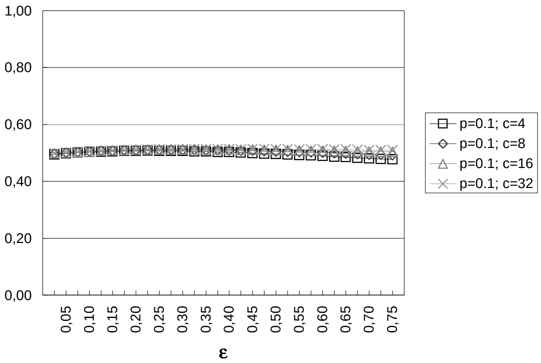


Figure B.125: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

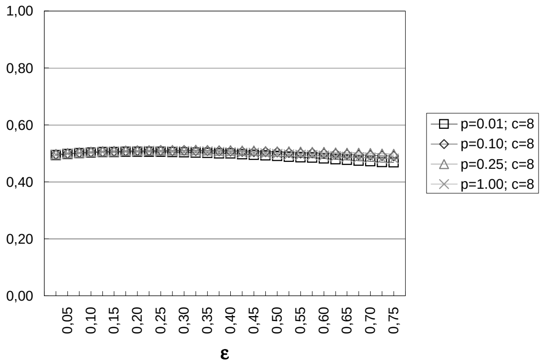


Figure B.126: Assignment problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

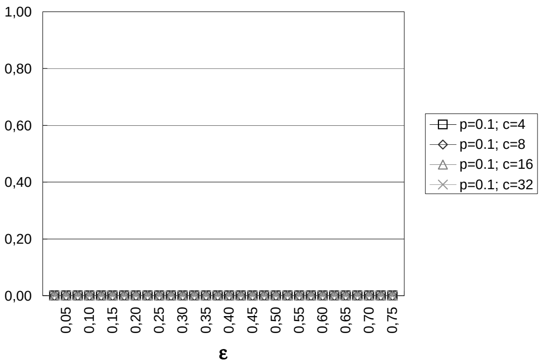


Figure B.127: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

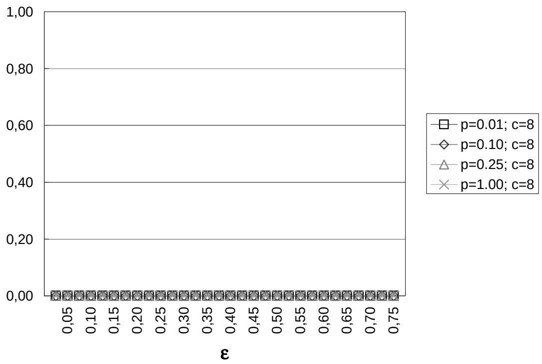


Figure B.128: Assignment problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

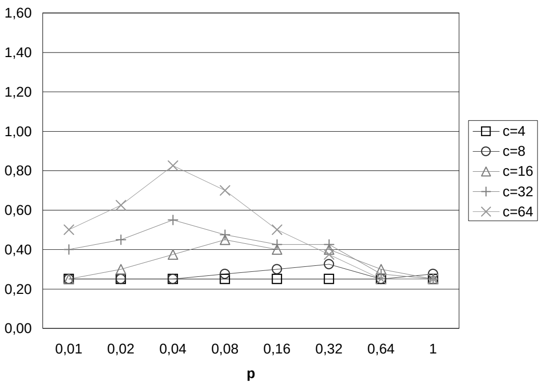


Figure B.129: Assignment problem – optimal penalty parameters $\varepsilon^m_*(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

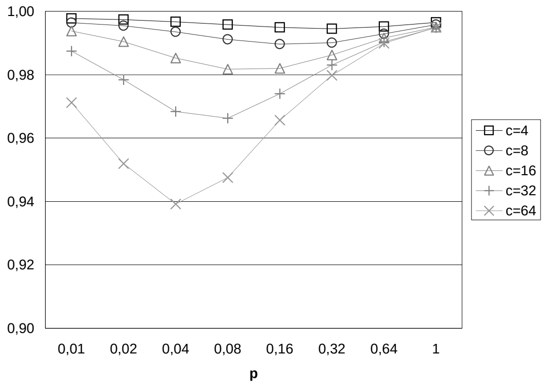


Figure B.130: Assignment problem – optimal average cost rates $\bar{\chi}^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

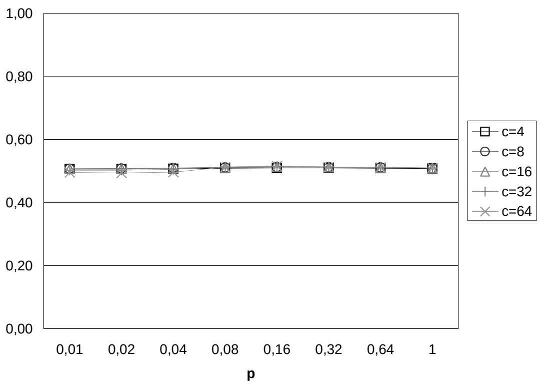


Figure B.131: Assignment problem – relative part of simulation runs $r^{<}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon^m_*)}), \widehat{w}(\check{S}_{2(\varepsilon^m_*)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

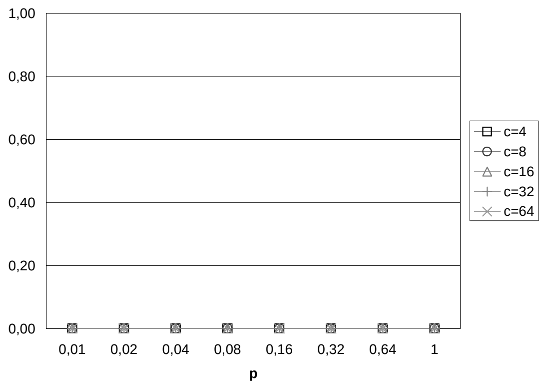


Figure B.132: Assignment problem – relative part of simulation runs $r^{=}_{\varepsilon^m_*}(p,c)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon^m_*)}), \widehat{w}(\check{S}_{2(\varepsilon^m_*)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

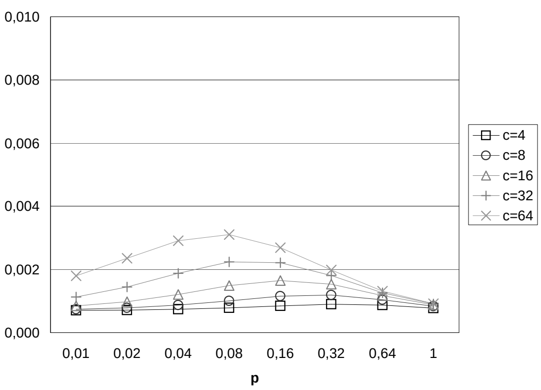


Figure B.133: Assignment problem – 98% certain maximal error $\check{\delta}^m_{\varepsilon_*}(p,c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

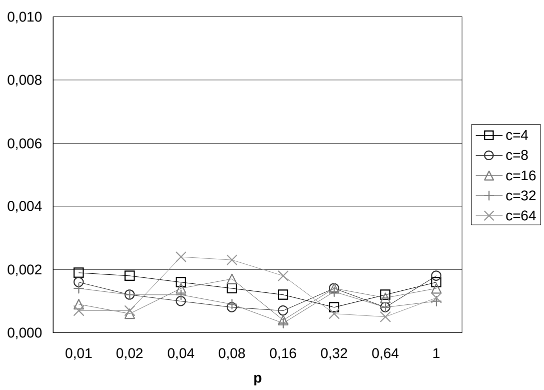


Figure B.134: Assignment problem – loss of quality $\triangle\bar{\chi}^m_{0.1}(p,c)$ by overestimating ε^m_* by 0.1; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

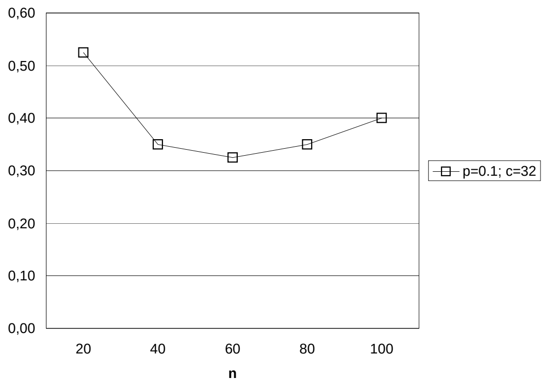


Figure B.135: Assignment problem – optimal penalty parameters $\varepsilon_*^m(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

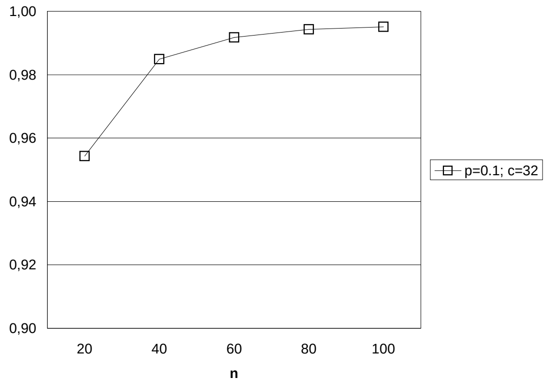


Figure B.136: Assignment problem – optimal average cost rates $\bar{\chi}_{\varepsilon_*}^m(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

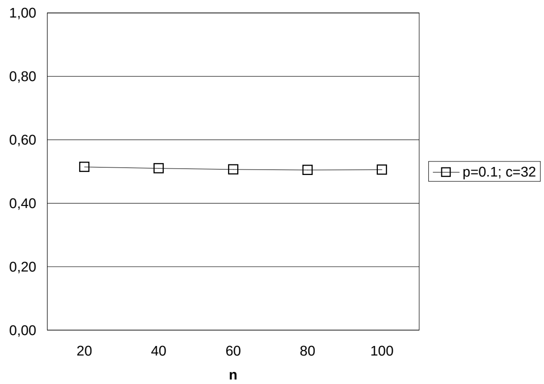


Figure B.137: Assignment problem – relative part of simulation runs $r_{\varepsilon_*^m}^<(n)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon_*^m)}), \widehat{w}(\check{S}_{2(\varepsilon_*^m)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

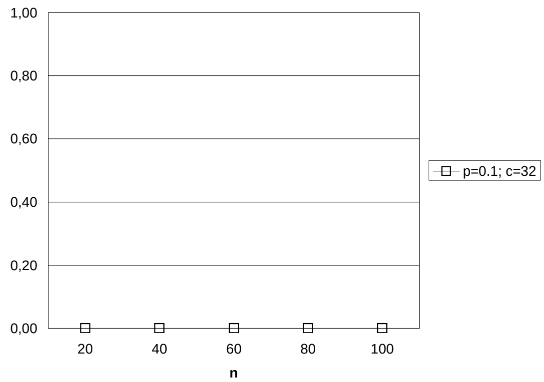


Figure B.138: Assignment problem – relative part of simulation runs $r_{\varepsilon_*^m}^=(n)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon_*^m)}), \widehat{w}(\check{S}_{2(\varepsilon_*^m)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

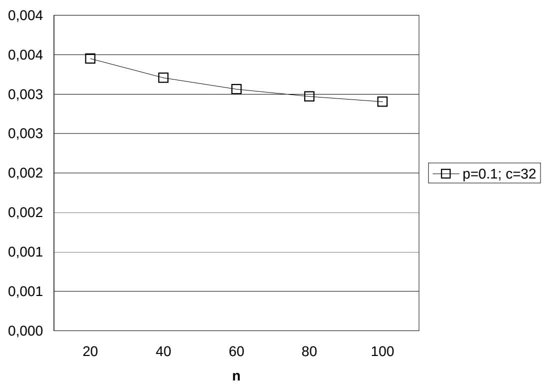


Figure B.139: Assignment problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}^m(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

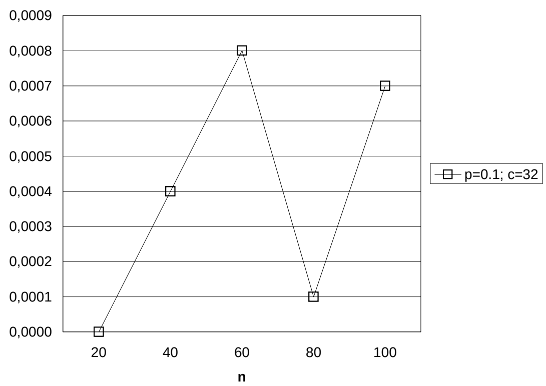


Figure B.140: Assignment problem – loss of quality $\triangle\bar{\chi}_{0.1}^m(n)$ by overestimating ε_*^m by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

B.2.2.2 Traveling Salesman Problem

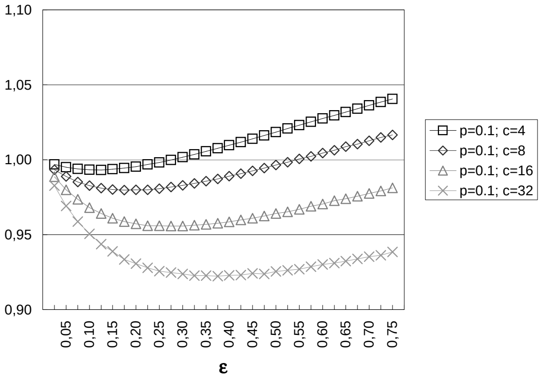


Figure B.141: Traveling salesman problem – average time rates $\bar{\chi}_{\varepsilon}^{m}$ for different ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

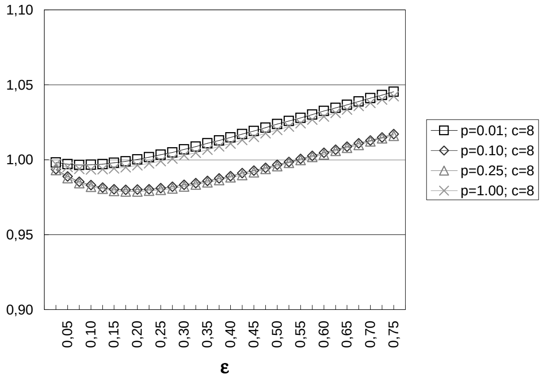


Figure B.142: Traveling salesman problem – average time rates $\bar{\chi}_{\varepsilon}^{m}$ for different ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

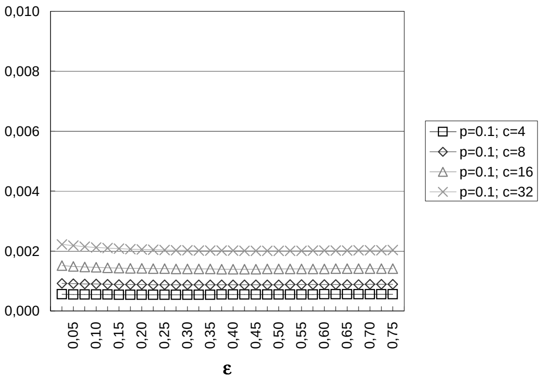


Figure B.143: Traveling salesman problem – 98% certain maximal error $\check{\delta}_{\varepsilon}^{m}$; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

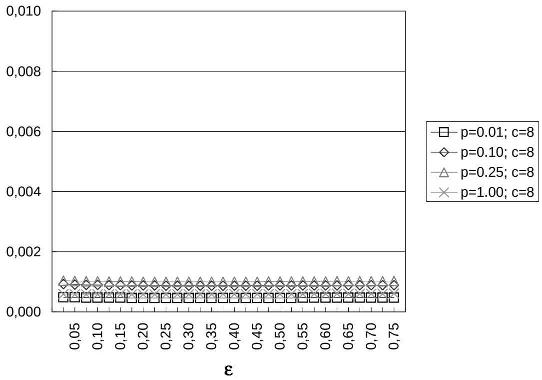


Figure B.144: Traveling salesman problem – 98% certain maximal error $\check{\delta}_{\varepsilon}^{m}$; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

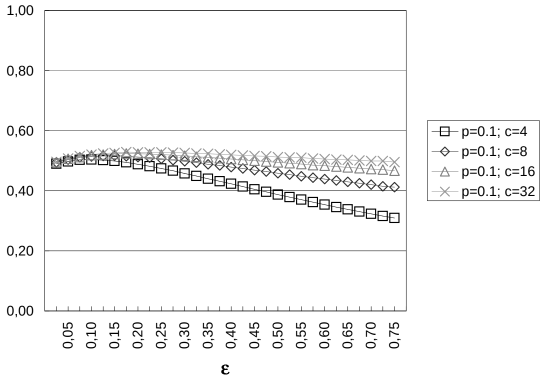


Figure B.145: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

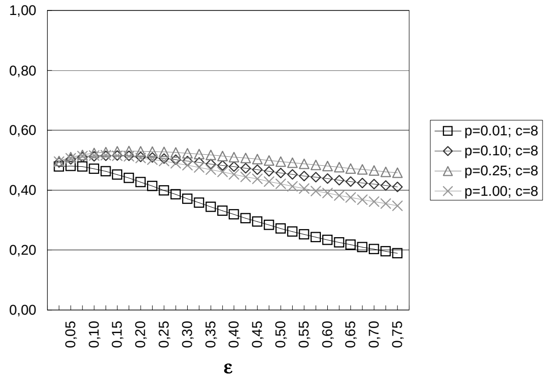


Figure B.146: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^<$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

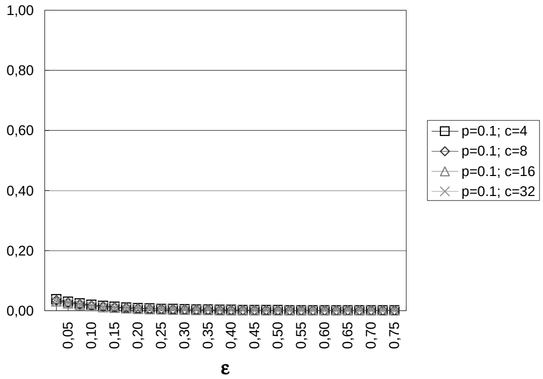


Figure B.147: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

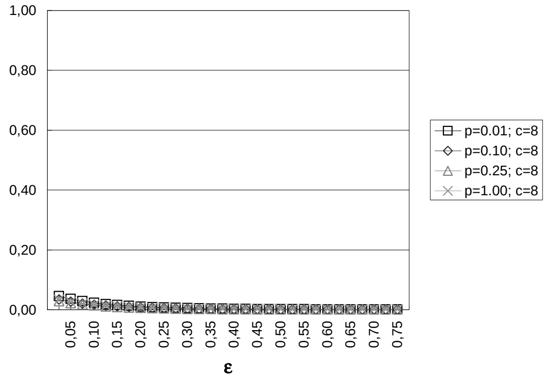


Figure B.148: Traveling salesman problem – relative part of simulation runs $r_\varepsilon^=$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon)}), \widehat{w}(\check{S}_{2(\varepsilon)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

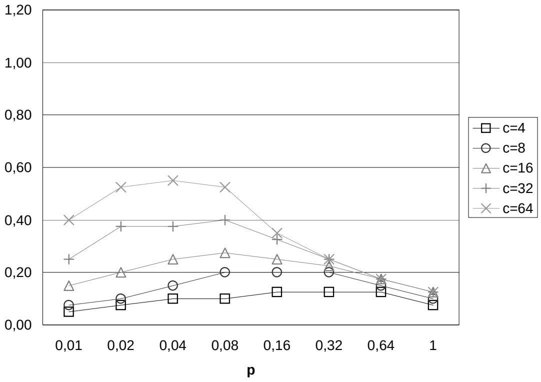


Figure B.149: Traveling salesman problem – optimal penalty parameters $\varepsilon_*^m(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

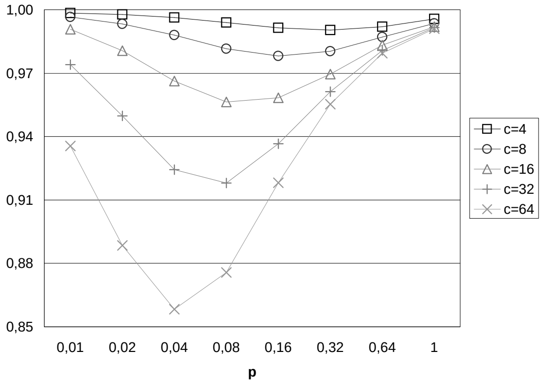


Figure B.150: Traveling salesman problem – optimal average time rates $\bar{\chi}_{\varepsilon_*}^m(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

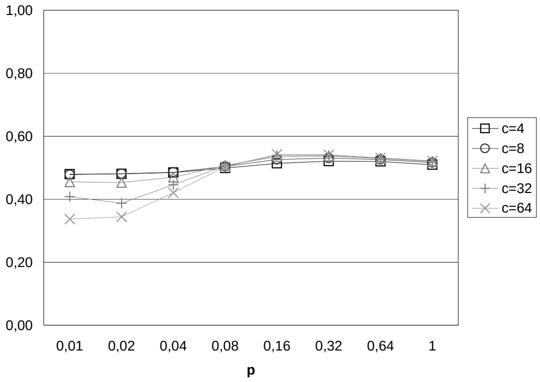


Figure B.151: Traveling salesman problem – relative part of simulation runs $r_{\varepsilon_*^m}^{<}(p, c)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon_*^m)}), \widehat{w}(\check{S}_{2(\varepsilon_*^m)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

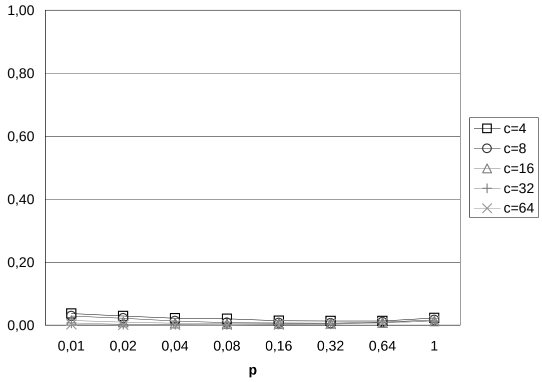


Figure B.152: Traveling salesman problem – relative part of simulation runs $r_{\varepsilon_*^m}^{=}(p, c)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon_*^m)}), \widehat{w}(\check{S}_{2(\varepsilon_*^m)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

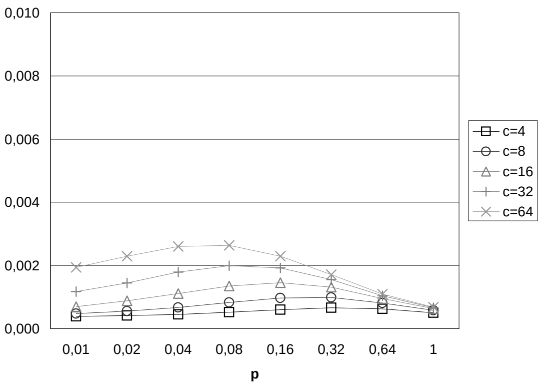


Figure B.153: Traveling salesman problem – 98% certain maximal error $\check{\delta}_{\varepsilon_*}^m(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

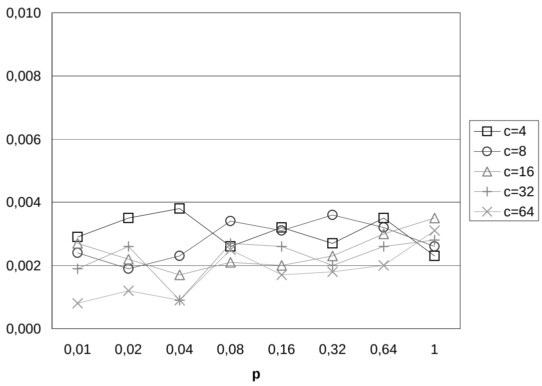


Figure B.154: Traveling salesman problem – loss of quality $\triangle\bar{\chi}_{0.1}^m(p, c)$ by overestimating ε_*^m by 0.1; $[I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

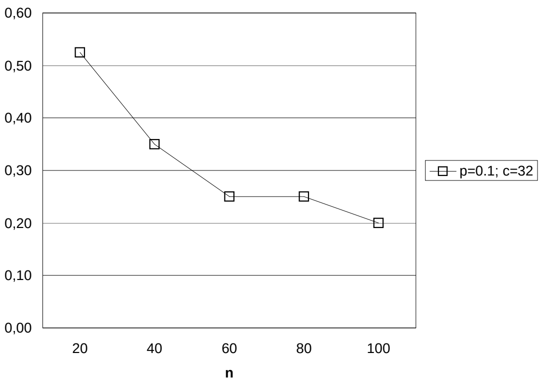


Figure B.155: Traveling salesman problem – optimal penalty parameters $\varepsilon^m_*(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

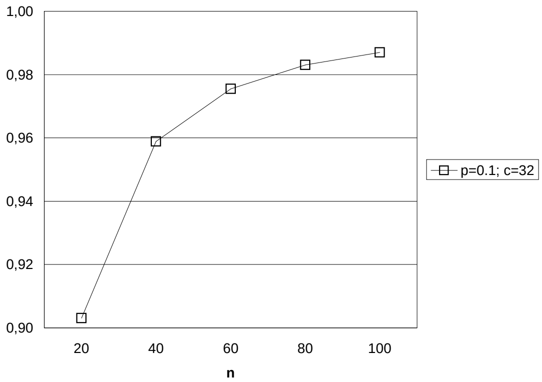


Figure B.156: Traveling salesman problem – optimal average time rates $\bar{\chi}^m_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

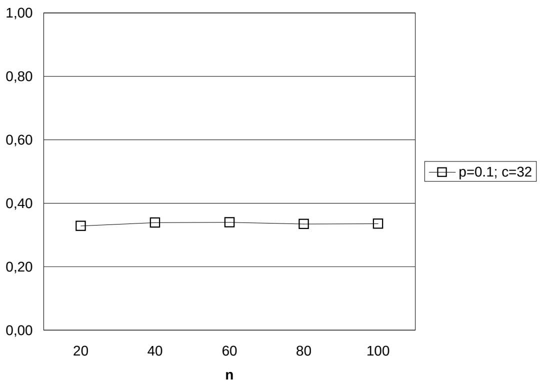


Figure B.157: Traveling salesman problem – relative part of simulation runs $r^{<}_{\varepsilon^m_*}(n)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon^m_*)}), \widehat{w}(\check{S}_{2(\varepsilon^m_*)})) < \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

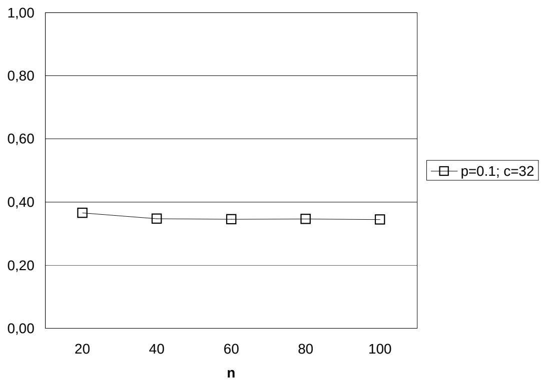


Figure B.158: Traveling salesman problem – relative part of simulation runs $r^{=}_{\varepsilon^m_*}(n)$ with $\min(\widehat{w}(\check{S}_{1(\varepsilon^m_*)}), \widehat{w}(\check{S}_{2(\varepsilon^m_*)})) = \min(\widehat{w}(\check{S}_{l1}), \widehat{w}(\check{S}_{l2}))$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

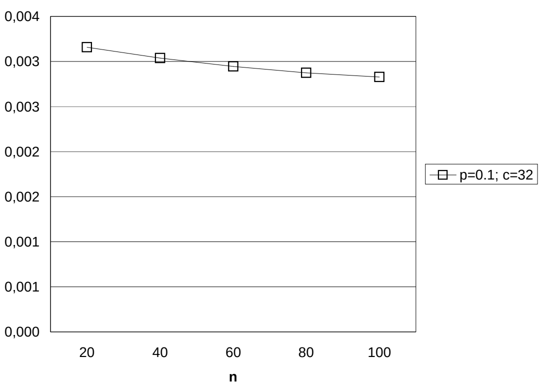


Figure B.159: Traveling salesman problem – 98% certain maximal error $\check{\delta}^m_{\varepsilon_*}(n)$; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

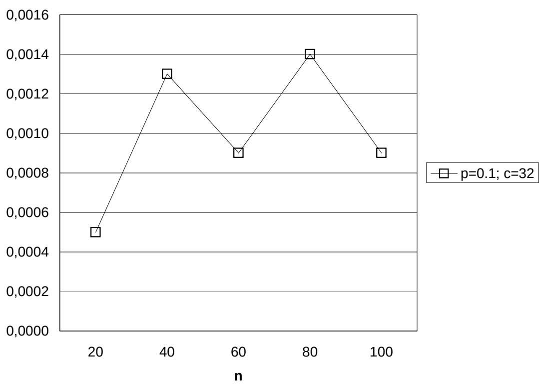


Figure B.160: Traveling salesman problem – loss of quality $\triangle\bar{\chi}^m_{0.1}(n)$ by overestimating ε^m_* by 0.1; $[I_\varepsilon = \{0.025, 0.050, \ldots, 1.50\};\ T = 10^6]$

Appendix C

Results – $(p; 1/c, c)$-Model

C.1 Exact Algorithms

C.1.1 Penalty Method (see Section 4.2)

C.1.1.1 Shortest Path Problem

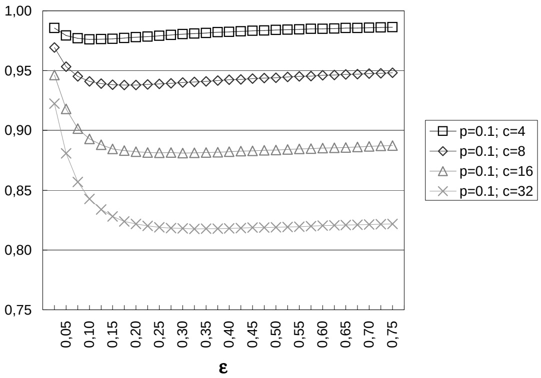


Figure C.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

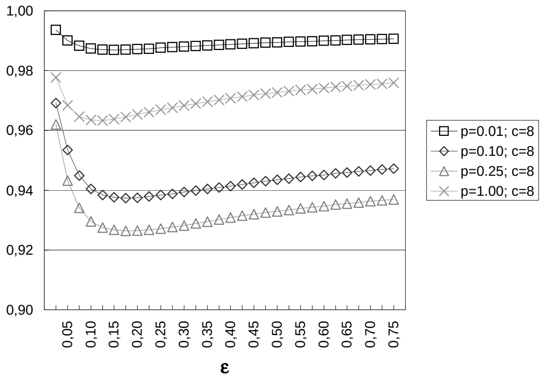


Figure C.2: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

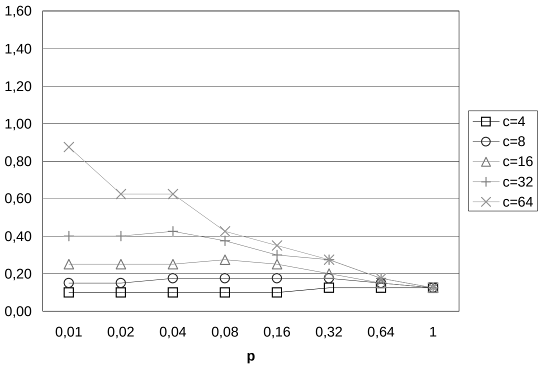


Figure C.3: Shortest path problem – optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

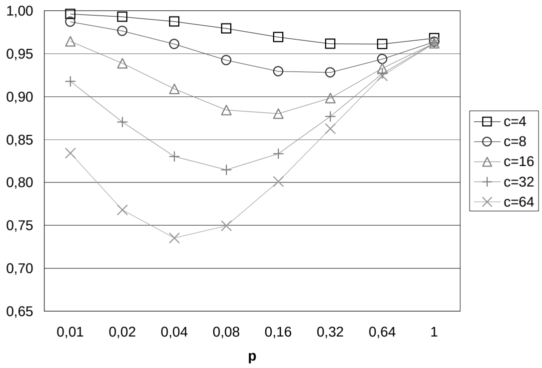


Figure C.4: Shortest path problem – optimal average time rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

C.1.1.2 Assignment Problem

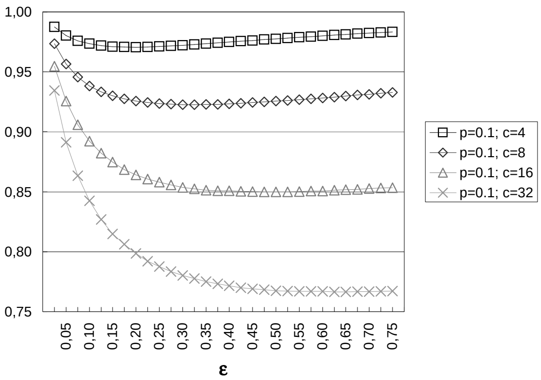


Figure C.5: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^5]$

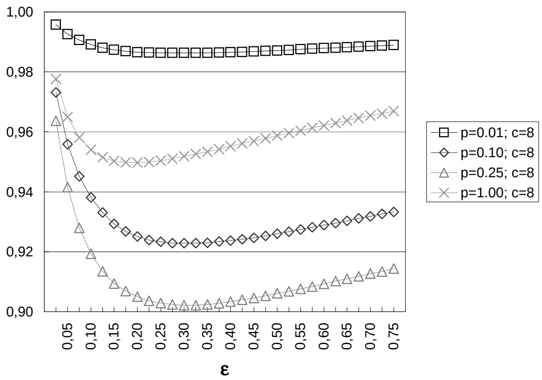


Figure C.6: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^5]$

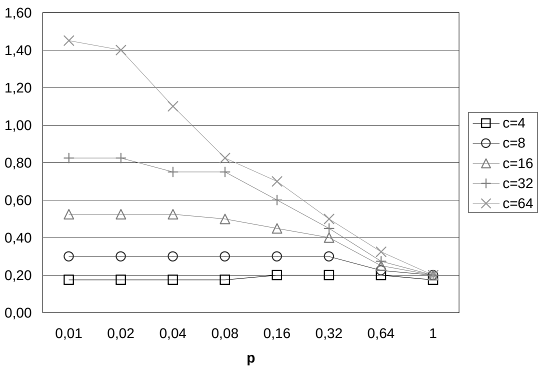


Figure C.7: Assignment problem – optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

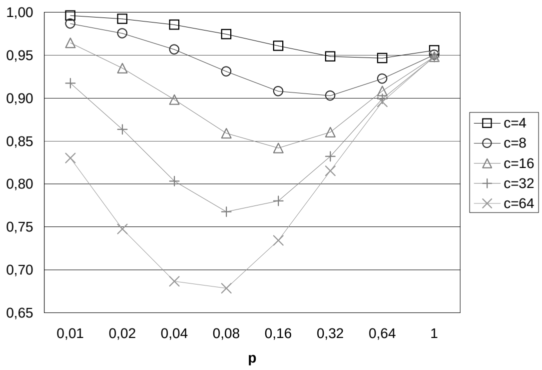


Figure C.8: Assignment problem – optimal average cost rates $\bar{\varphi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^5]$

C.1.2 Mutual Penalty Method (see Section 4.2)

C.1.2.1 Shortest Path Problem

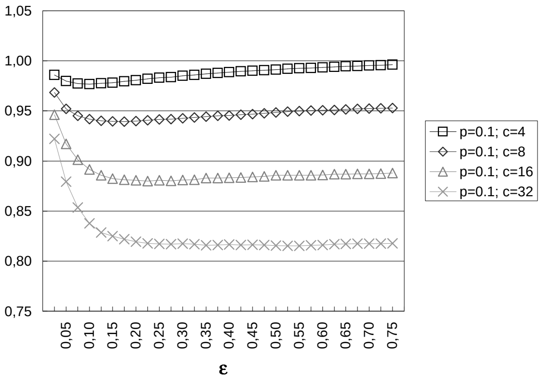


Figure C.9: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

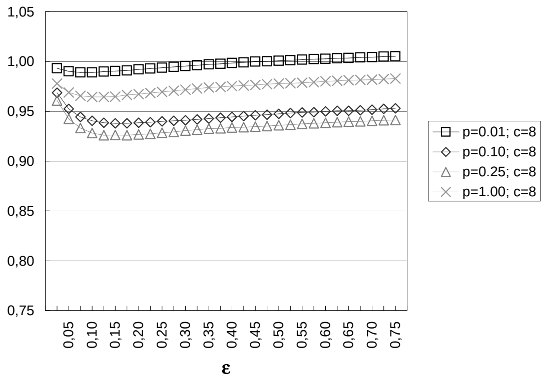


Figure C.10: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

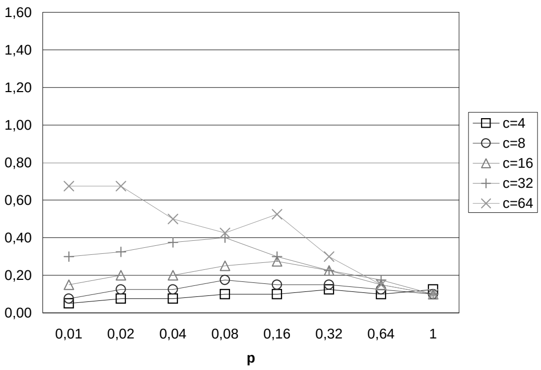


Figure C.11: Shortest path problem – optimal penalty parameters $\varepsilon_{*}^{m}(p, c)$; $[n = 25;\ I_{\varepsilon} = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

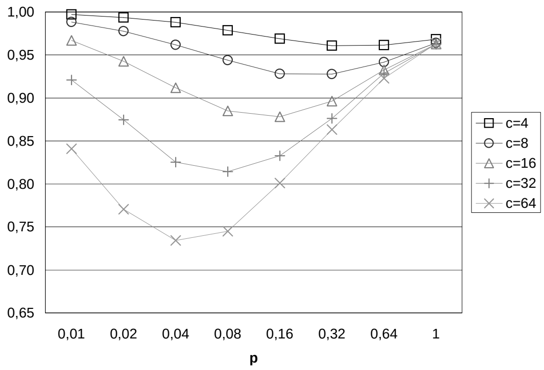


Figure C.12: Shortest path problem – optimal average time rates $\bar{\varphi}_{\varepsilon_*}^{m}(p, c)$; $[n = 25;\ I_{\varepsilon} = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

C.1.2.2 Assignment Problem

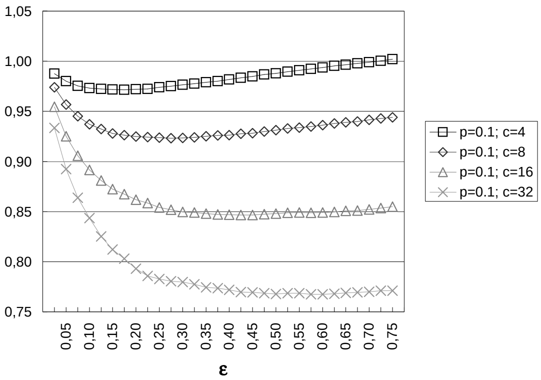


Figure C.13: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^4]$

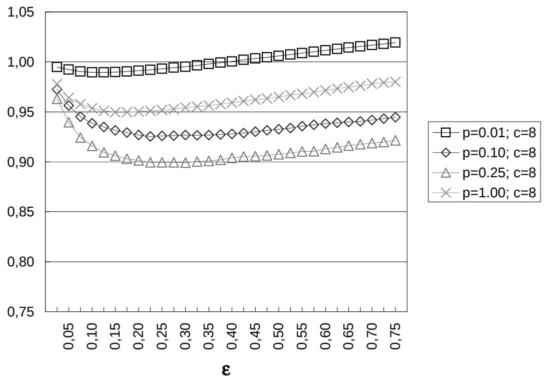


Figure C.14: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^4]$

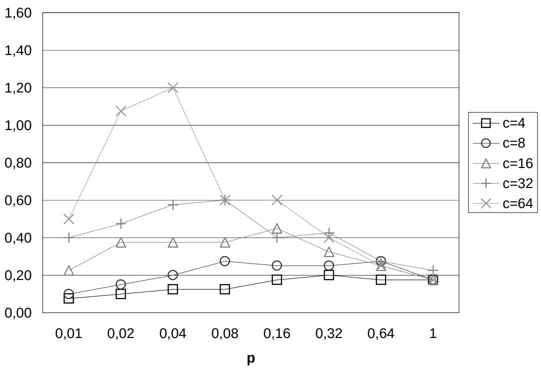


Figure C.15: Assignment problem – optimal penalty parameters $\varepsilon_{*}^{m}(p, c)$; $[n = 25;\ I_{\varepsilon} = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

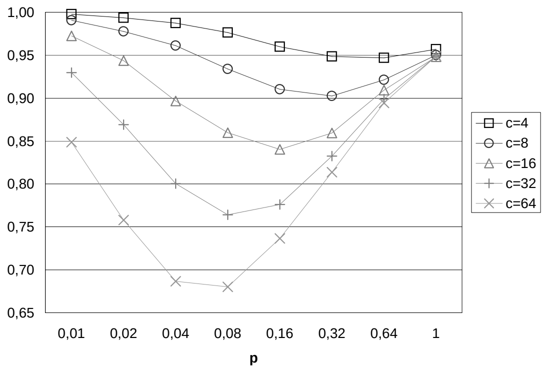


Figure C.16: Assignment problem – optimal average cost rates $\bar{\varphi}_{\varepsilon_*}^{m}(p, c)$; $[n = 25;\ I_{\varepsilon} = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^4]$

C.2 Heuristic Algorithms

C.2.1 Penalty Method (see Section 4.2)

C.2.1.1 Assignment Problem

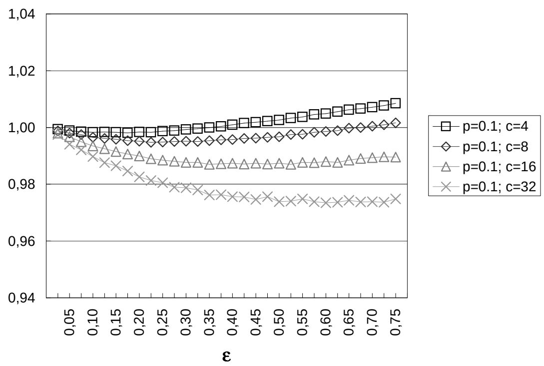


Figure C.17: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

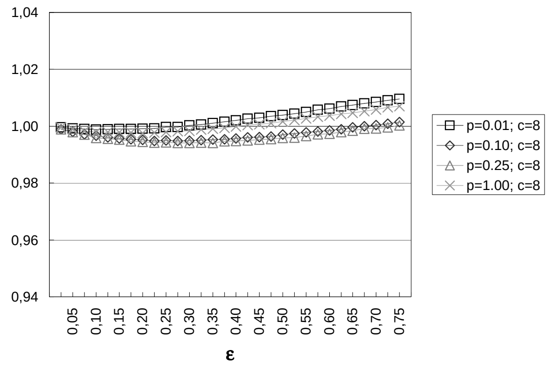


Figure C.18: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

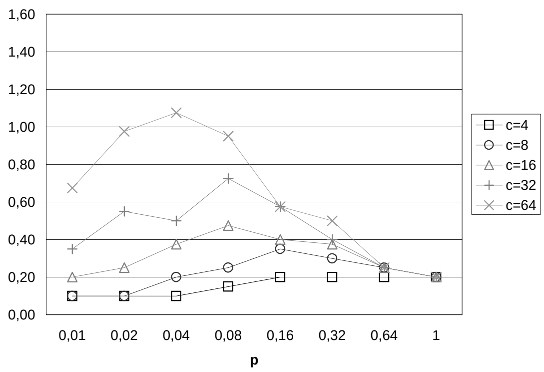


Figure C.19: Assignment problem – optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

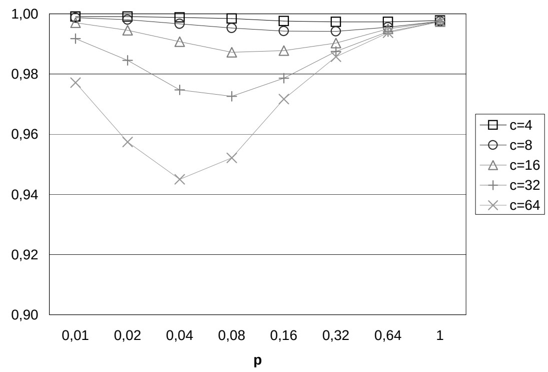


Figure C.20: Assignment problem – optimal average cost rates $\bar{\chi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

C.2.1.2 Traveling Salesman Problem

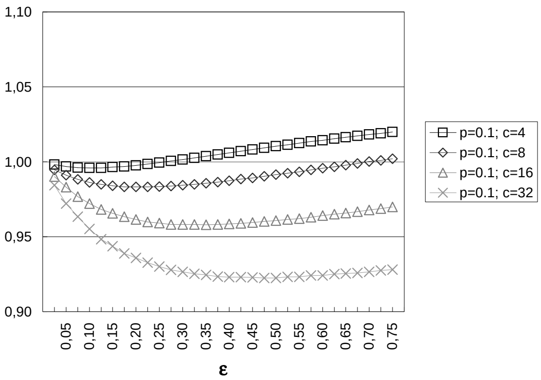


Figure C.21: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

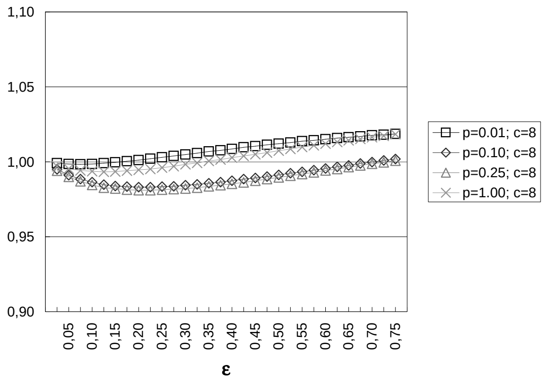


Figure C.22: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

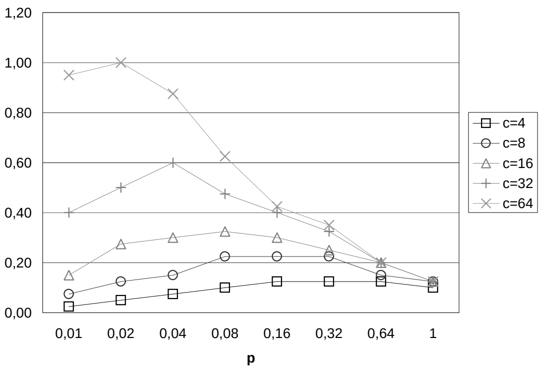


Figure C.23: Traveling salesman problem – optimal penalty parameters $\varepsilon_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

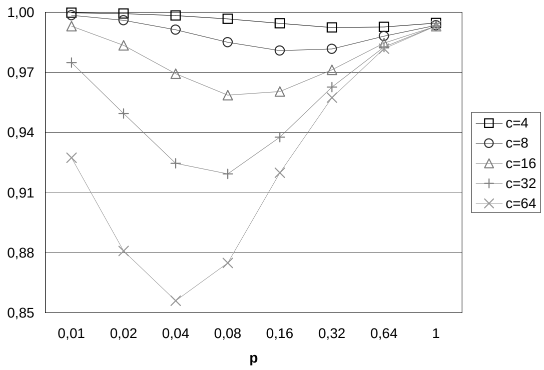


Figure C.24: Traveling salesman problem – optimal average time rates $\bar{\chi}_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

C.2.2 Mutual Penalty Method (see Section 4.2)

C.2.2.1 Assignment Problem

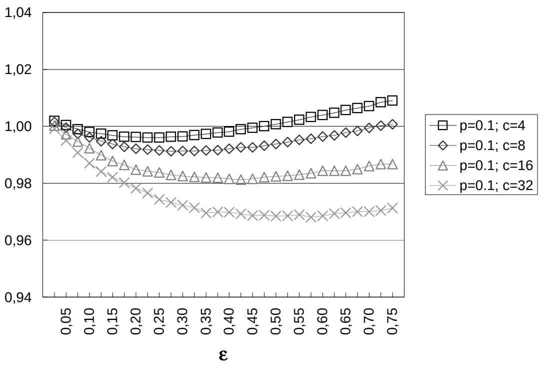


Figure C.25: Assignment problem – average cost rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ p = 0.1;\ c = 4, 8, 16, 32;\ T = 10^6]$

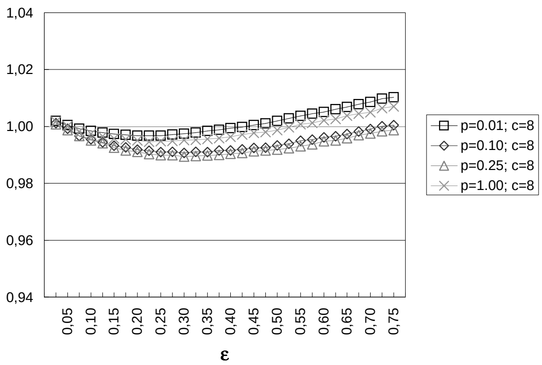


Figure C.26: Assignment problem – average cost rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ p = 0.01, 0.1, 0.25, 1.00;\ c = 8;\ T = 10^6]$

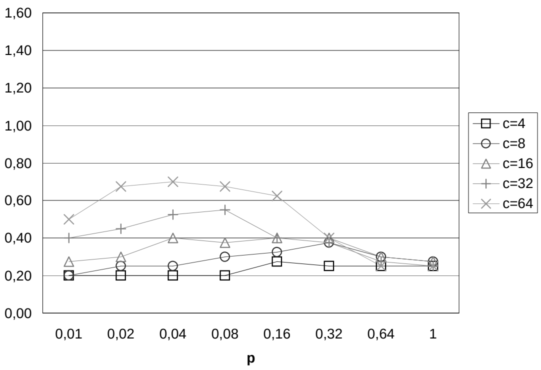


Figure C.27: Assignment problem – optimal penalty parameters $\varepsilon^m_*(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

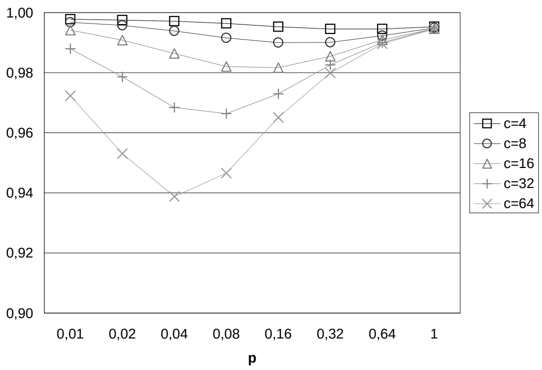


Figure C.28: Assignment problem – optimal average cost rates $\bar{\varphi}^m_{\varepsilon_*}(p, c)$; $[n = 25;\ I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\};\ T = 10^6]$

C.2.2.2 Traveling Salesman Problem

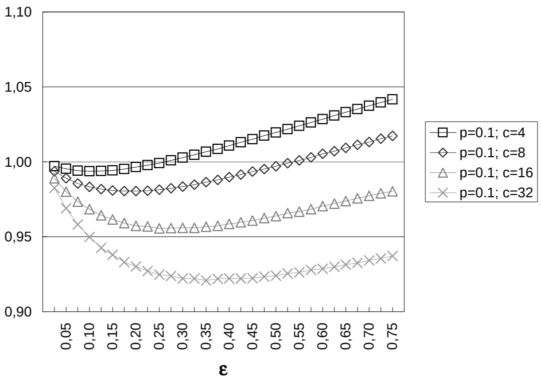


Figure C.29: Traveling salesman problem – average time rates $\bar{\chi}^m_\varepsilon$ for different ε; [$n = 25$; $p = 0.1$; $c = 4, 8, 16, 32$; $T = 10^6$]

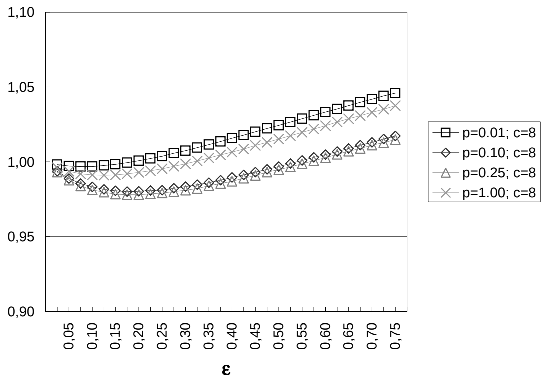


Figure C.30: Traveling salesman problem – average time rates $\bar{\chi}^m_\varepsilon$ for different ε; [$n = 25$; $p = 0.01, 0.1, 0.25, 1.00$; $c = 8$; $T = 10^6$]

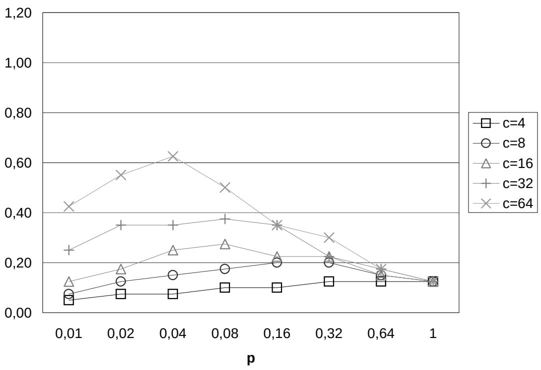


Figure C.31: Traveling salesman problem – optimal penalty parameters $\varepsilon^m_*(p, c)$; [$n = 25$; $I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\}$; $T = 10^6$]

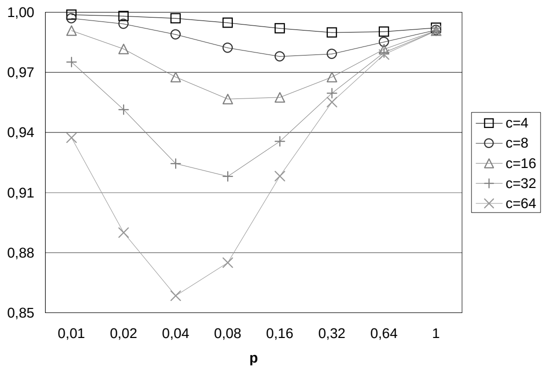


Figure C.32: Traveling salesman problem – optimal average time rates $\bar{\chi}^m_{\varepsilon_*}(p, c)$; [$n = 25$; $I_\varepsilon = \{0.025, 0.05, \ldots, 2.50\}$; $T = 10^6$]

Appendix D

Results – $(p := 1 - w(e); 1, c)$-Model

D.1 Exact Algorithms

D.1.1 Penalty Method (see Section 4.2)

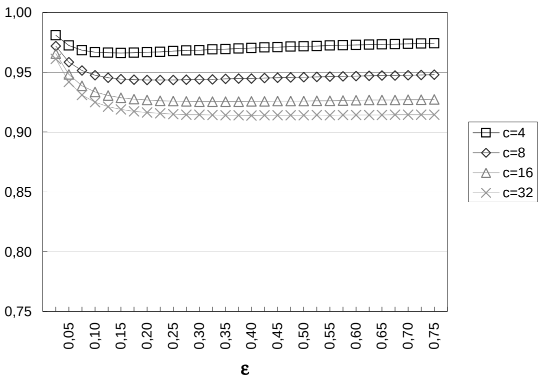


Figure D.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

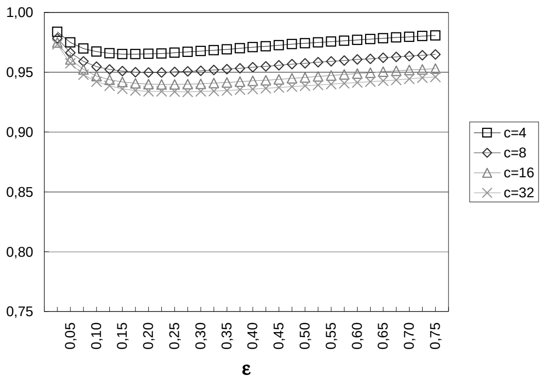


Figure D.2: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

	c	2	4	8	16	32	64
SP	ε_*	0.075	0.150	0.225	0.275	0.375	0.625
	$\bar{\varphi}_{\varepsilon_*}$	0.988	0.966	0.944	0.925	0.914	0.907
ASP	ε_*	0.125	0.175	0.200	0.225	0.250	0.275
	$\bar{\varphi}_{\varepsilon_*}$	0.985	0.965	0.949	0.939	0.933	0.931

Table D.1: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average time/cost rates $\bar{\varphi}_{\varepsilon_*}(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^5]$

D.1.2 Mutual Penalty Method (see Section 4.2)

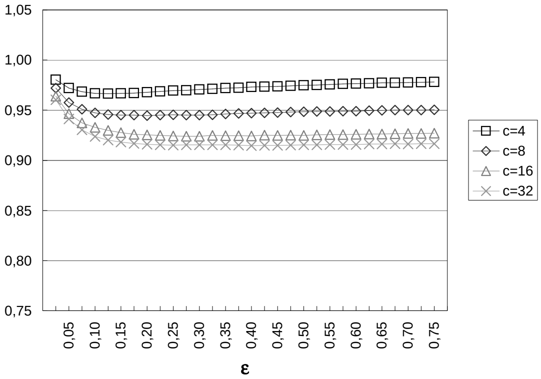


Figure D.3: Shortest path problem – average time rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

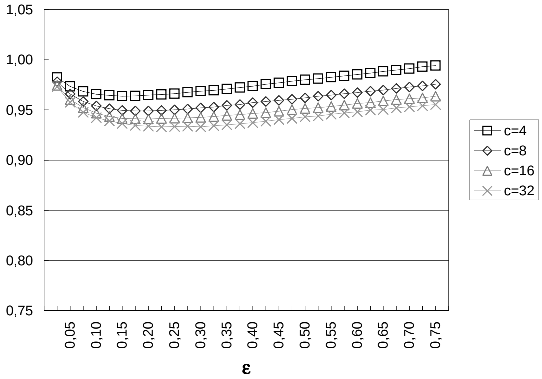


Figure D.4: Assignment problem – average cost rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

	c	2	4	8	16	32	64
SP	ε^m_*	0.075	0.150	0.275	0.325	0.400	0.350
	$\bar{\varphi}^m_{\varepsilon_*}$	0.988	0.966	0.942	0.925	0.914	0.906
ASP	ε^m_*	0.100	0.150	0.175	0.225	0.200	0.225
	$\bar{\varphi}^m_{\varepsilon_*}$	0.985	0.965	0.949	0.939	0.934	0.931

Table D.2: The optimal penalty parameters $\varepsilon^m_*(c)$ and the optimal average time/cost rates $\bar{\varphi}^m_{\varepsilon_*}(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^4]$

D.2 Heuristic Algorithms

D.2.1 Penalty Method (see Section 4.2)

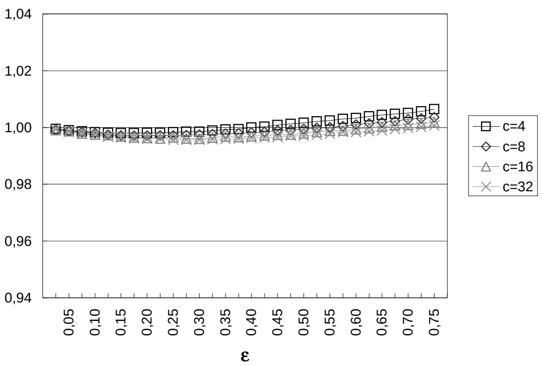


Figure D.5: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

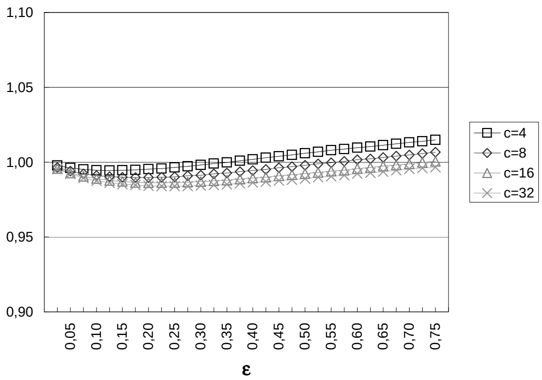


Figure D.6: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

	c	2	4	8	16	32	64
ASP	ε_*	0.150	0.150	0.200	0.250	0.275	0.250
	$\bar{\chi}_{\varepsilon_*}$	0.998	0.997	0.997	0.995	0.995	0.995
TSP	ε_*	0.075	0.125	0.200	0.225	0.225	0.250
	$\bar{\chi}_{\varepsilon_*}$	0.998	0.994	0.990	0.986	0.984	0.983

Table D.3: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average cost/time rates $\bar{\chi}_{\varepsilon_*}(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^6]$

D.2.2 Mutual Penalty Method (see Section 4.2)

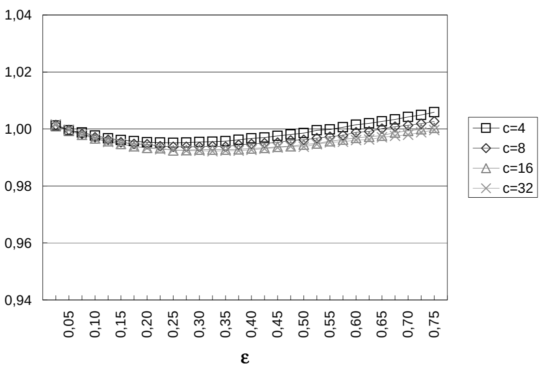


Figure D.7: Assignment problem – average cost rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

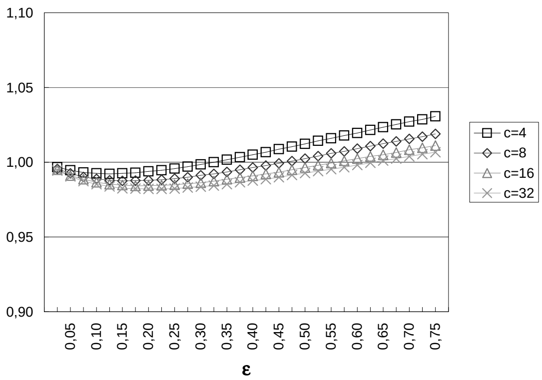


Figure D.8: Traveling salesman problem – average time rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

	c	2	4	8	16	32	64
ASP	ε^m_*	0.225	0.250	0.300	0.325	0.325	0.350
	$\bar{\chi}^m_{\varepsilon_*}$	0.997	0.995	0.994	0.993	0.992	0.991
TSP	ε^m_*	0.075	0.125	0.150	0.200	0.200	0.225
	$\bar{\chi}^m_{\varepsilon_*}$	0.997	0.992	0.987	0.984	0.982	0.981

Table D.4: The optimal penalty parameters $\varepsilon^m_*(c)$ and the optimal average cost/time rates $\bar{\chi}^m_{\varepsilon_*}(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^6]$

Appendix E

Results – $(p := \frac{1}{c+1}; 1, c)(1 - p; \frac{1}{c}, 1)$-Model

E.1 Exact Algorithms

E.1.1 Penalty Method (see Section 4.2)

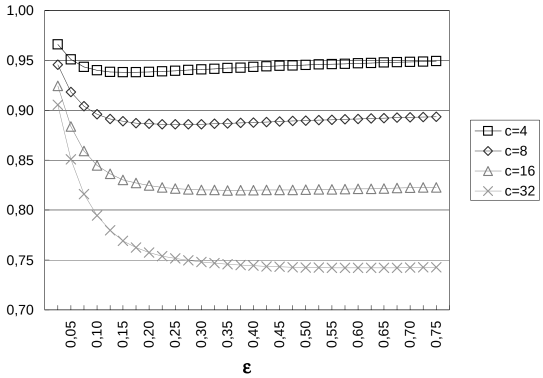


Figure E.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

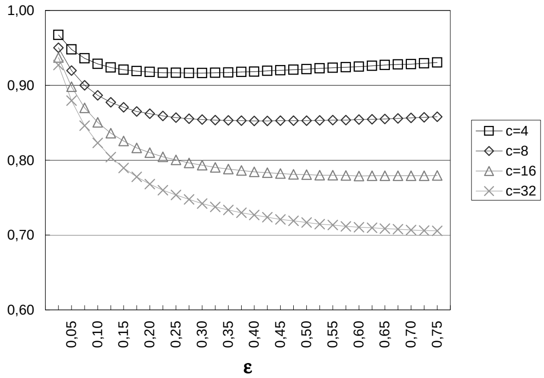


Figure E.2: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

	c	2	4	8	16	32	64
SP	ε_*	0.100	0.150	0.225	0.425	0.575	1.050
	$\bar{\varphi}_{\varepsilon_*}$	0.976	0.938	0.886	0.821	0.742	0.661
ASP	ε_*	0.150	0.250	0.425	0.700	1.000	1.500
	$\bar{\varphi}_{\varepsilon_*}$	0.967	0.916	0.853	0.779	0.700	0.630

Table E.1: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average time/cost rates $\bar{\varphi}_{\varepsilon_*}(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^5]$

E.1.2 Mutual Penalty Method (see Section 4.2)

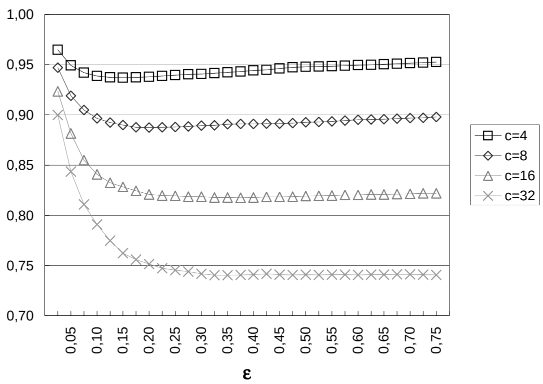


Figure E.3: Shortest path problem – average time rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

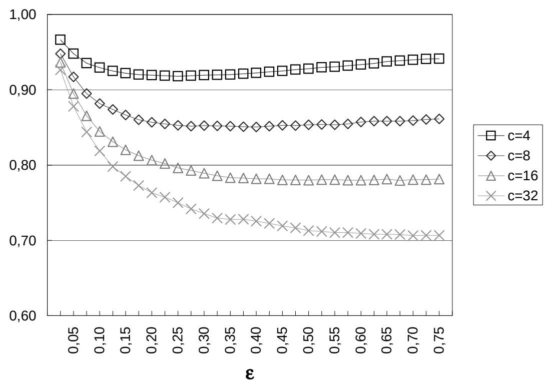


Figure E.4: Assignment problem – average cost rates $\bar{\varphi}^m_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

	c	2	4	8	16	32	64
SP	ε^m_*	0.075	0.150	0.250	0.325	0.525	0.725
	$\bar{\varphi}^m_{\varepsilon_*}$	0.976	0.937	0.886	0.824	0.735	0.655
ASP	ε^m_*	0.125	0.225	0.400	0.425	0.875	1.250
	$\bar{\varphi}^m_{\varepsilon_*}$	0.968	0.914	0.848	0.781	0.698	0.639

Table E.2: The optimal penalty parameters $\varepsilon^m_*(c)$ and the optimal average time/cost rates $\bar{\varphi}^m_{\varepsilon_*}(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^4]$

E.2 Heuristic Algorithms

E.2.1 Penalty Method (see Section 4.2)

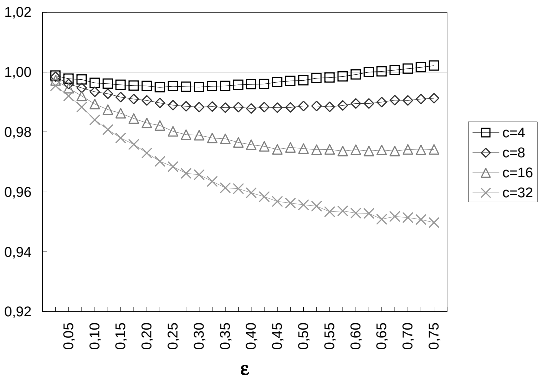


Figure E.5: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

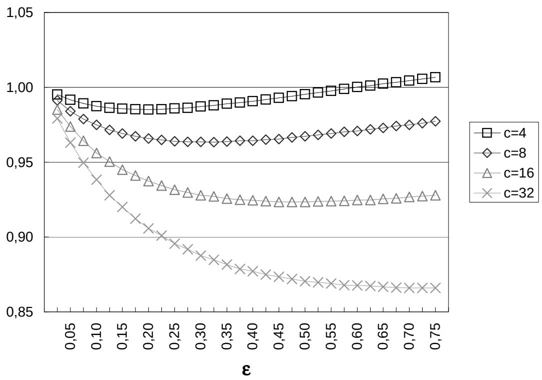


Figure E.6: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

	c	2	4	8	16	32	64
ASP	ε_*	0.150	0.200	0.375	0.600	1.000	1.450
	$\bar{\chi}_{\varepsilon_*}$	0.998	0.995	0.988	0.974	0.949	0.916
TSP	ε_*	0.100	0.175	0.325	0.525	0.775	1.250
	$\bar{\chi}_{\varepsilon_*}$	0.996	0.986	0.963	0.924	0.866	0.802

Table E.3: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average cost/time rates $\bar{\chi}_{\varepsilon_*}(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^6]$

E.2.2 Mutual Penalty Method (see Section 4.2)

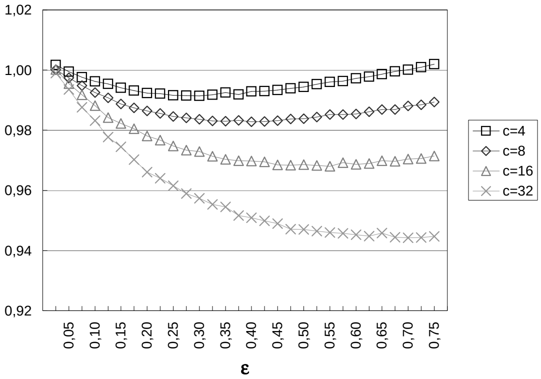


Figure E.7: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

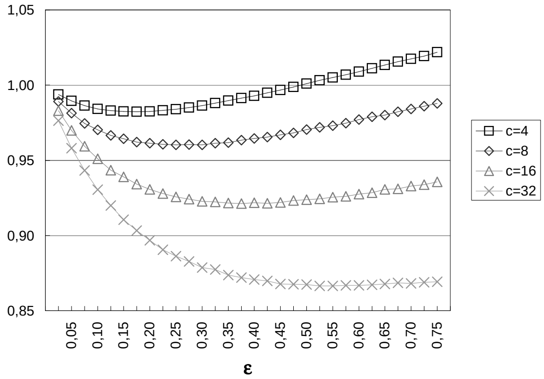


Figure E.8: Traveling salesman problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

	c	2	4	8	16	32	64
ASP	ε_*^m	0.225	0.275	0.400	0.475	0.675	0.975
	$\bar{\chi}_{\varepsilon_*}^m$	0.996	0.991	0.983	0.968	0.943	0.913
TSP	ε_*^m	0.100	0.175	0.275	0.375	0.575	0.750
	$\bar{\chi}_{\varepsilon_*}^m$	0.994	0.982	0.960	0.922	0.867	0.811

Table E.4: The optimal penalty parameters $\varepsilon_*^m(c)$ and the optimal average cost/time rates $\bar{\chi}_{\varepsilon_*}^m(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^6]$

Appendix F

Results – $OSN(1, c^2)$-Model

F.1 Exact Algorithms

F.1.1 Penalty Method (see Section 4.2)

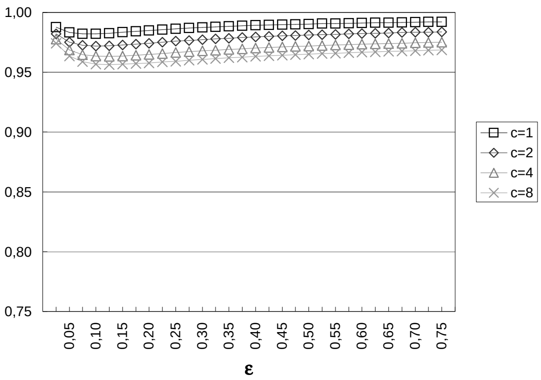


Figure F.1: Shortest path problem – average time rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

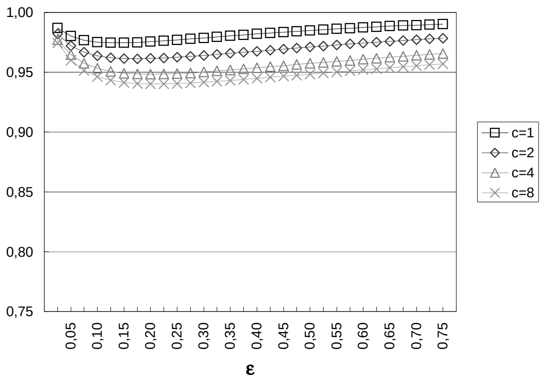


Figure F.2: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^5]$

	c	1	2	4	8	16	32
SP	ε_*	0.075	0.100	0.125	0.125	0.125	0.150
	$\bar{\varphi}_{\varepsilon_*}$	0.982	0.972	0.963	0.956	0.952	0.950
ASP	ε_*	0.125	0.175	0.200	0.225	0.225	0.225
	$\bar{\varphi}_{\varepsilon_*}$	0.975	0.961	0.949	0.939	0.935	0.931

Table F.1: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average time/cost rates $\bar{\varphi}_{\varepsilon_*}(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^5]$

F.1.2 Mutual Penalty Method (see Section 4.2)

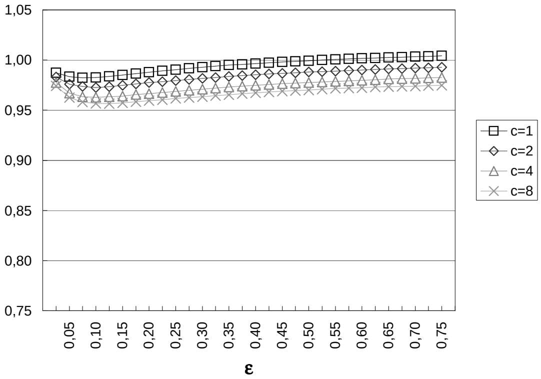


Figure F.3: Shortest path problem – average time rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

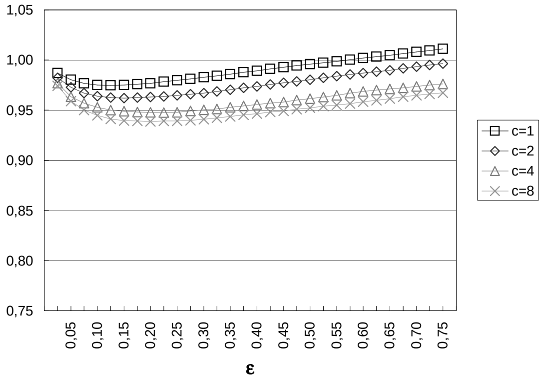


Figure F.4: Assignment problem – average cost rates $\bar{\varphi}_{\varepsilon}^{m}$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^4]$

	c	1	2	4	8	16	32
SP	ε_*^m	0.075	0.100	0.100	0.100	0.125	0.150
	$\bar{\varphi}_{\varepsilon_*}^m$	0.981	0.972	0.963	0.957	0.952	0.950
ASP	ε_*^m	0.125	0.150	0.175	0.200	0.200	0.200
	$\bar{\varphi}_{\varepsilon_*}^m$	0.975	0.961	0.948	0.939	0.934	0.932

Table F.2: The optimal penalty parameters $\varepsilon_*^m(c)$ and the optimal average time/cost rates $\bar{\varphi}_{\varepsilon_*}^m(c)$ for the shortest path problem (SP) and the assignment problem (ASP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^4]$

F.2 Heuristic Algorithms

F.2.1 Penalty Method (see Section 4.2)

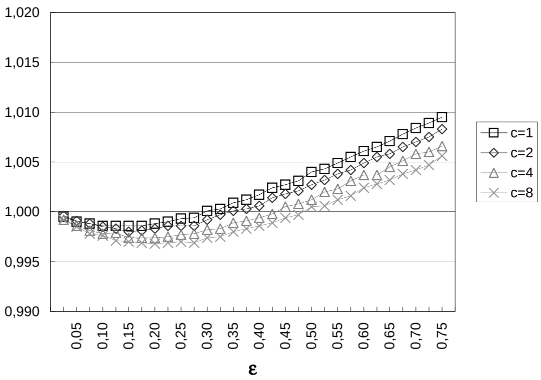


Figure F.5: Assignment problem – average cost rates $\bar{\chi}_\varepsilon$ for different ε; [$n = 25$; $c = 4, 8, 16, 32$; $T = 10^6$]

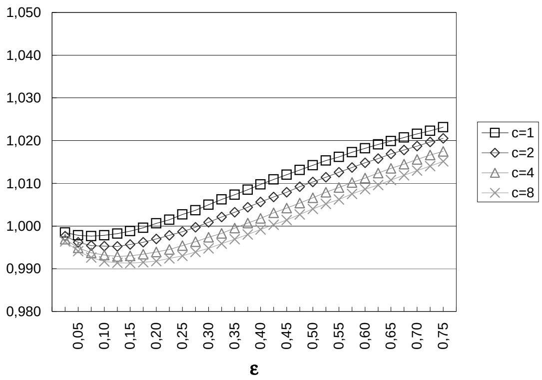


Figure F.6: Traveling salesman problem – average time rates $\bar{\chi}_\varepsilon$ for different ε; [$n = 25$; $c = 4, 8, 16, 32$; $T = 10^6$]

	c	1	2	4	8	16	32
ASP	ε_*	0.125	0.150	0.150	0.200	0.225	0.225
	$\bar{\chi}_{\varepsilon_*}$	0.998	0.997	0.997	0.997	0.996	0.996
TSP	ε_*	0.075	0.100	0.125	0.150	0.175	0.175
	$\bar{\chi}_{\varepsilon_*}$	0.998	0.995	0.993	0.991	0.990	0.989

Table F.3: The optimal penalty parameters $\varepsilon_*(c)$ and the optimal average cost/time rates $\bar{\chi}_{\varepsilon_*}(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); [$n = 25$; $c = 2, 4, 8, 16, 32, 64$; $T = 10^6$]

F.2.2 Mutual Penalty Method (see Section 4.2)

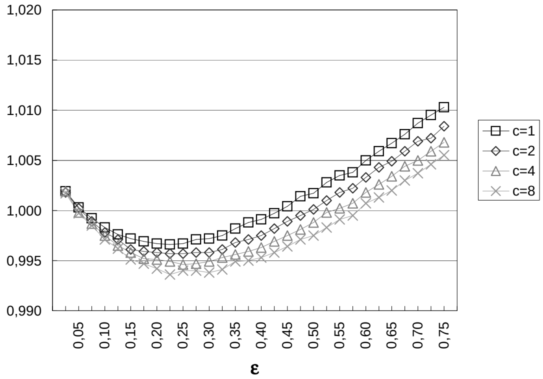


Figure F.7: Assignment problem – average cost rates $\bar{\varphi}_\varepsilon^m$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

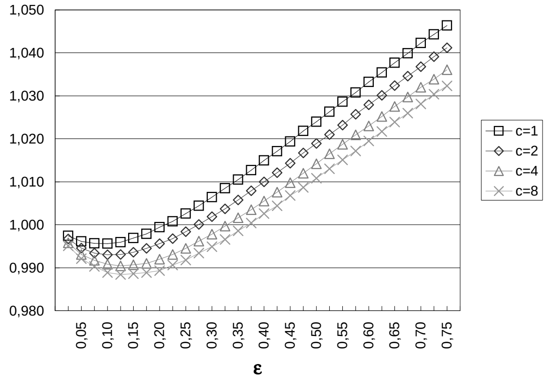


Figure F.8: Traveling salesman problem – average time rates $\bar{\varphi}_\varepsilon^m$ for different ε; $[n = 25;\ c = 4, 8, 16, 32;\ T = 10^6]$

	c	1	2	4	8	16	32
ASP	ε_*^m	0.200	0.200	0.200	0.275	0.250	0.300
	$\bar{\chi}_{\varepsilon_*}^m$	0.997	0.996	0.995	0.994	0.993	0.993
TSP	ε_*^m	0.075	0.100	0.125	0.150	0.150	0.150
	$\bar{\chi}_{\varepsilon_*}^m$	0.995	0.993	0.990	0.989	0.988	0.987

Table F.4: The optimal penalty parameters $\varepsilon_*^m(c)$ and the optimal average cost/time rates $\bar{\chi}_{\varepsilon_*}^m(c)$ for the assignment problem (ASP) and the traveling salesman problem (TSP); $[n = 25;\ c = 2, 4, 8, 16, 32, 64;\ T = 10^6]$

3-Hirn-Verlag

ISBN 3-937583-06-8