MATERIAL FLOW COST VERSUS CONGESTION IN DYNAMIC DISTRIBUTED FACILITY LAYOUT PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYKUT ÖZEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JUNE 2008

Approval of the thesis:

MATERIAL FLOW COST VERSUS CONGESTION IN DYNAMIC DISTRIBUTED FACILITY LAYOUT PROBLEM

Submitted by **AYKUT ÖZEN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Canan Özgen ____________________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Nur Evin Özdemirel ____________________
Head of Department, **Industrial Engineering**

Prof. Nur Evin Özdemirel ____________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Canan Sepil ____________________
Industrial Engineering Dept., METU

Prof. Nur Evin Özdemirel ____________________
Industrial Engineering Dept., METU

Prof. Nesim Erkip ____________________
Industrial Engineering Dept., Bilkent University

Assoc. Prof. Haldun Süral ____________________
Industrial Engineering Dept., METU

Assist. Prof. Müge Avşar ____________________
Industrial Engineering Dept., METU

Date: 06.12.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

MATERIAL FLOW COST VERSUS CONGESTION IN
DYNAMIC DISTRIBUTED FACILITY LAYOUT PROBLEM

Özen, Aykut
M.S., Department of Industrial Engineering
Supervisor : Prof. Dr. Nur Evin Özdemirel

June 2008, 109 pages

In this thesis, we study both dynamic and distributed facility layout problems, where the demand for product mix changes over time. We propose a new simulated annealing algorithm, SALAB, for the dynamic facility layout problem. Four variants of SALAB find the best known solution for 20 of the 48 benchmark problems from the literature, improving upon the best known solutions of 18 problems. We modify SALAB to obtain DSALAB, solving the dynamic distributed facility layout problem with the objective of minimizing relocation cost and total (full and empty) travel cost of the material handling system. We simulate DSALAB solutions of randomly generated problems to study the tradeoff between total cost and congestion in the system. Our experimental results indicate that distributing the department duplicates throughout the facility reduces the total cost with diminishing returns and causes increasing congestion. Therefore, distribution beyond a certain level is not justified.

Keywords: Dynamic Distributed Facility Layout Problem, Simulated Annealing, Material Flow cost, Congestion, Work-in-Process

ÖZ

DİNAMİK VE DAĞITIK TESİS YERLEŞİM PROBLEMİNDE
MALZEME AKIŞ MALİYETİ VE SIKIŞIKLIK

Özen, Aykut
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Nur Evin Özdemirel

Haziran 2008, 109 sayfa

Bu tezde, ürün yelpazesine olan talebin zamanla değiştiği dinamik ve dağıtık tesis yerleşim problemleri ele alınmıştır. Dinamik tesis yerleşim problemi için yeni bir tavlama benzetimi algoritması – SALAB – geliştirilmiştir. SALAB'ın dört farklı versiyonu, literatürden alınan 48 test probleminin 20'si için bilinen en iyi sonuçları bulurken 18 problem için bilinenden daha iyi sonuç bulmuştur. Makinaların taşınma maliyetini ve malzeme taşıma sistemine ait toplam (boş ve dolu) hareket maliyetini azaltma hedefiyle, SALAB uyarlanarak, dinamik ve dağıtık tesis yerleşim problemini çözen DSALAB geliştirilmiştir. Sistemdeki toplam maliyet ve sıkışıklık arasındaki ödünleşimi incelemek için, rastsal üretilen problemlerin DSALAB çözümleri simüle edilmiştir. Deneysel sonuçlarımıza göre makina kopyalarını tesis bütününde dağıtmak toplam maliyeti azalan getirilerle düşürmekte ve artan sıkışıklığa sebep olmaktadır. Dolayısıyla, belirli bir seviyenin ötesinde dağıtıklık tavsiye edilmemektedir.

Anahtar Kelimeler : Dinamik ve Dağıtık Tesis Yerleşim Problemi, Tavlama Benzetimi, Malzeme Akış Maliyeti, Sıkışıklık, Ara Stok

To My Parents and Güzide

ACKNOWLEDGEMENTS

The author wishes to express his deepest gratitude to his supervisor Prof. Dr. Nur Evin Özdemirel. Her advices and criticism throughout the research were crucial.

The author would also like to thank the members of the examining committee for their valuable comments and critiques.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Facility layout design directly affects the operational performance of a production environment. Carrying a huge amount of material to distant locations in a facility can decrease the operational performance in terms of material handling costs, resource utilizations, Work-In-Process inventory (WIP), production lead times, and so on. These issues should be taken into account while developing a layout plan in which the locations of departments or machines are specified.

Facility layout problems generally aim at minimizing total cost of interdepartmental material flow weighted with distance, considering area constraints to construct layout plans. The Static Facility Layout Problem (SFLP) deals with placement of departments considering the material handling costs. Balakrishnan and Cheng (1998) define SFLP as follows: “given a group of departments, the material flow between each pair of departments, and the cost per unit of flow per unit distance, the departments have to be arranged into a layout such that the sum of the costs of flow between the departments in the layout is minimized”. Material flow between each pair of departments is represented by an aggregate flow matrix (over all product types) for the planning horizon. However, this material flow can be deterministic or stochastic. Moreover, it can change over time periods because of the changes in product mixes, product routings and production amounts. These changes lead to replacement of departments and thus to the Dynamic Facility Layout Problem (DFLP).

Based on their work in 20 companies, Benjaafar et al. (2002) conclude that “companies value layouts that retain their usefulness over many product mixes or can easily be reconfigured. Equally important are layouts that permit shorter lead times, lower inventories, and a greater degree of product customization”. Conventional layouts, which are “typically designed for a specific product mix and production volume that are assumed to continue for a sufficiently long period”, do not meet these needs.

In addition to the conventional distance-weighted material flow cost criterion, there are different models that consider other performance measures such as the empty (unloaded) trips of material handling system (MHS), resource utilizations, WIP and production lead times. These approaches usually use queuing network formulations to approximately find the values of performance measures. However, these studies are limited with single period problems (SFLP).

In a functional layout, machines of the same type are grouped to lie together. Benjaafar and Sheikhzadeh (2000) and Lahmar and Benjaafar (2005) propose to distribute those machines (by disaggregating large departments into small subdepartments) throughout the layout in order to improve the layout performance. By this way, product types can have alternate routes by visiting different duplicates of the same machine type distributed throughout the facility. Benjaafar and Sheikhzadeh (2000) and Lahmar and Benjaafar (2005) show that such distributed layouts are efficient in terms of material flow costs. However, it is unknown how this duplication affects congestion and other performance measures of the production system.

In this thesis, we focus on the deterministic layout problems with equal sized departments or machines. Specifically, we study the Dynamic Distributed Facility Layout Problem (DDFLP), which is a more general form of DFLP that also takes distributed layout into consideration. DFLP decides on the location of each department in each time period by considering material handling and relocation costs. It responds to flow changes by relocating machines, which incurs the relocation cost. In addition to relocation, DDFLP also has the mechanism of distributing duplicate machines throughout the facility. Distribution provides additional flexibility and reduces material handling cost. In addition to deciding on the location of each machine duplicate in each time period, we also need to determine the flow allocations. That is, different part types that need a certain machine type should be routed to one of the duplicates of that machine type. Hence, if the flow allocations in DDFLP are known, the problem reduces to DFLP.

Before finding a solution strategy for DDFLP, solution methods for DFLP have been reviewed. According to the results of McKendall et al. (2006), Simulated Annealing (SA) heuristic with a look ahead/back strategy is a promising method for DFLP. In this study, SA algorithm of McKendall et al. (2006) is slightly changed by using a different neighborhood definition. After setting the annealing parameters, this algorithm named SALAB is tested on benchmark problems of Balakrishnan and Cheng (2000). Some solutions of the algorithm are found to be better than the best known results. Layout problems generally minimize the total cost of material flow between machines. This means that only the full (loaded) trips of material handling system are taken into account and empty (unloaded) trips are ignored. Our DDFLP objective function considers total material handling cost (both full and empty trips of material handling system). DDFLP concurrently decides on the location of machines and flow allocation among duplicates of a machine type. Therefore flow allocation decision is added to SALAB to solve DDFLP. In the final algorithm, cost of both full and empty trips of the material handling system and machine relocation cost are considered in the objective. This algorithm is used to improve random initial solutions for varying duplication levels and dynamically changing demand characteristics.

Minimizing material handling cost does not guarantee reducing congestion in the system (Benjaafar, 2002). Therefore, we simulate the SA solutions to estimate the resource utilization and WIP levels. By this way, effects of varying duplication levels and demand characteristics on congestion are experimentally analyzed.

The contribution of this thesis can be summarized as follows.

- SA algorithm of McKendall et al. (2006) for DFLP is modified and improved upon.
- Empty travel cost of the material handling system is included in the objective function of DDFLP. The SA algorithm is adapted to solve DDFLP including minimization of total material handling cost and relocation cost in the objective.
- A simulation model is developed for DDFLP and used to explore the operational performance of SA solutions for different department duplication levels and demand characteristics.

The thesis is organized as follows. DFLP and DDFLP formulation and solution methods are reviewed in Chapter 2. Alternative layout formulations considering additional operational performance metrics are also discussed in this chapter. Our SA algorithm for DFLP, SALAB, is described and compared with other solution approaches in Chapter 3. In Chapter 4, we give our DDFLP formulation, which is based on the formulation by Lahmar and Benjaafar (2005). We then describe adaptation of SALAB to solve DDFLP. The simulation model for predicting operational performance of DDFLP solutions and its integration with the SA algorithm is also explained in this chapter. Chapter 5 includes problem generation, experimentation scheme and results. Finally, we conclude with Chapter 6.

CHAPTER 2

LITERATURE REVIEW

According to a recent review by Drira et al. (2007), layout problems can be studied with varying levels of detail and concerns. A detailed representation of facility layout problem characteristics given in Drira et al. (2007) is shown in Figure 2.1. Key issues of modeling a layout problem are representation form and detail (equal or unequal sized blocks, multi or single floor, and so on), objective function (material handling cost, re-layout cost, work-in-process inventory, flow times, and so on) and material flow characteristics (deterministic or stochastic, static or dynamic). From the perspective of layout evolution, Drira et al. (2007) classifies layout problems as static and dynamic. These two classes correspond to conventional layouts (typically manufacturing system layouts in Figure 2.1) and recently emerging layouts with a dynamic component. We briefly review these classes below.

Conventional Layouts

According to Drira et al. (2007), layout design generally depends on "the product variety and the production volumes". Four conventional organizations are fixed product, process, product and cellular layouts. Fixed product layout is used for immobile products such as ships. Instead of the product, production resources are moved to perform the operations on the product. Process layout (functional departments) is used when there are a wide variety of products. Product layout is used for high production volumes with low variety of products. Cellular layout is used for grouping machines into cells for production of a family of products.

Recently Emerging Layouts

In their literature review on facility layout, Benjaafar et al. (2002) classify the layout types according to two levels of production uncertainty and two levels of relocation cost (relocation cost of departments over consecutive time periods) as in Table 2.1. For the low

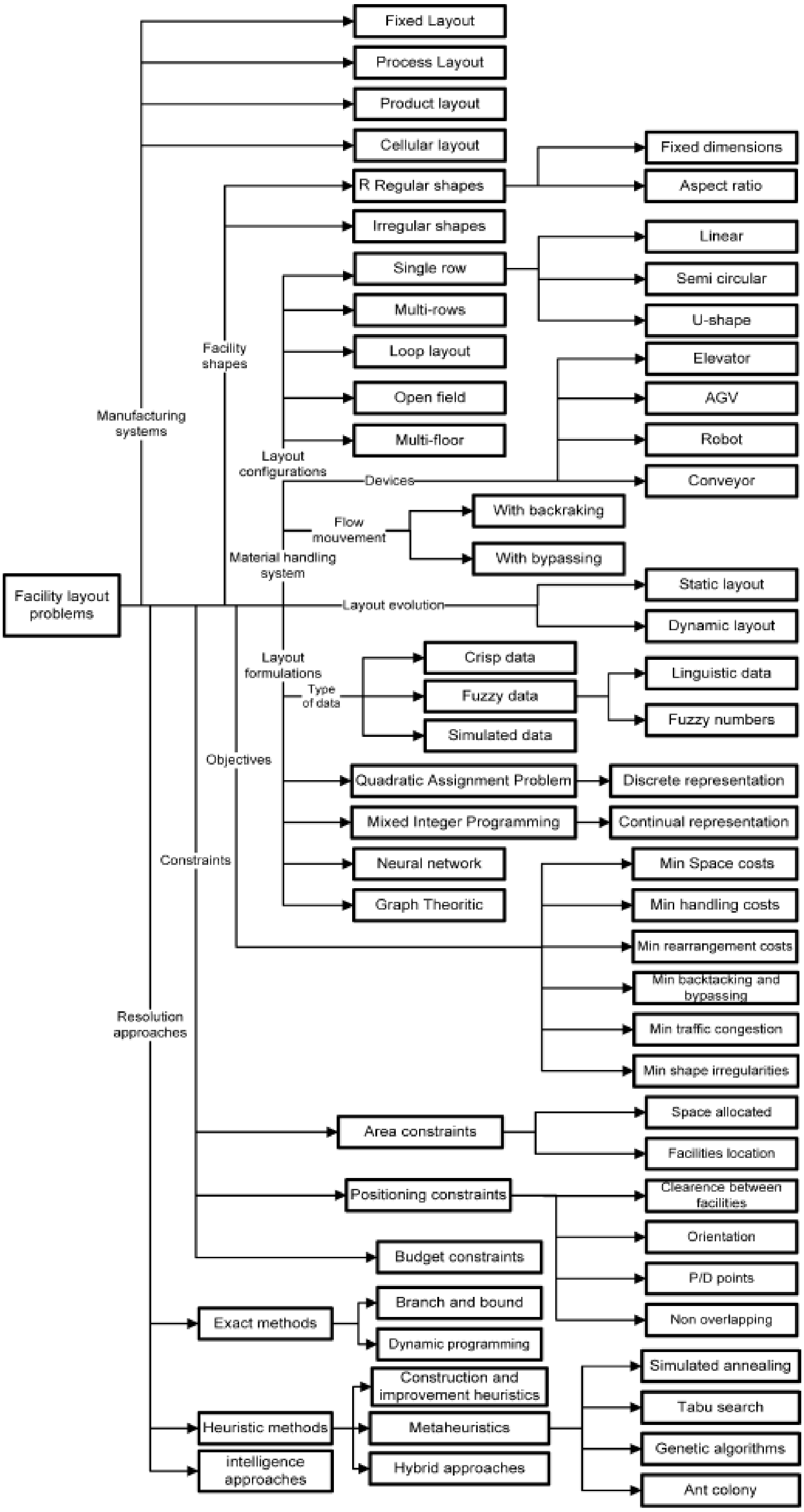


Figure 2.1. Tree representation of the layout problems (Drira et al., 2007)

Table 2.1. Layouts for relocation cost vs. demand uncertainty (Benjaafar et al., 2002)

	Uncertainty of Future Production Requirements	
Cost of re-layout	Low	High
Low	Dynamic layout	Reconfigurable layout
High	Robust layout	Distributed layout

relocation cost cases, layout can be changed easily in each period to minimize the material handling cost of that period. It is assumed that material handling costs dominate the relocation costs for this case, and the two choices are dynamic layout and reconfigurable layout. For the high relocation cost cases, relocation costs dominate the material handling costs. Hence, relatively stable layouts (robust and distributed) should be chosen for such cases.

In the recent survey paper, Kulturel-Konak (2007) state that there are two approaches to designing robust and/or flexible facilities. The first approach is the Dynamic Facility Layout Problem (DFLP) which is a multi period problem in which the material flow in each period is deterministic and known. “Facility layout arrangements are determined for each period by balancing material handling costs with the relayout costs involved in changing the layout between periods”. Dynamic facility layout problems are modeled for equal or unequal sized departments. According to Kulturel-Konak (2007), the second approach is Stochastic Facility Layout Problem (StoFLP) in which product mix and demand are assumed to be random variables with known parameters. It is stated that “most stochastic FLP research focuses on two important notions: *flexibility for future changes* and *robustness to uncertainty*. A robust facility is one that behaves well over a variety of scenarios and outcomes. On the other hand, a flexible facility is one that can readily adapt to changes without significantly affecting performance”. Flexibility, dominance (optimality) and robustness approaches of StoFLP and solution approaches for DFLP from the literature are discussed in the survey paper.

In their survey, Balakrishnan and Cheng (1998) classify the algorithms of DFLP for equal and unequal sized departments. Equal sized department case is also categorized as having deterministic or stochastic material flow.

Their work on 20 companies has led Benjaafar et al. (2002) to the conclusion that “companies value layouts that retain their usefulness over many product mixes or can easily be reconfigured. Equally important are layouts that permit shorter lead times, lower inventories, and a greater degree of product customization”. Conventional layouts which are “typically designed for a specific product mix and production volume that are assumed to continue for a sufficiently long period” do not meet these needs. Moreover, they say that “three approaches to layout design address three distinct needs of the flexible factory”. Those approaches are “distributed, modular and agile layouts”. Distributed layouts distribute the department duplicates (by disaggregating a large department) throughout the floor. Modular layouts are a hybrid modular form of functional, flow line and cellular types. Agile layouts try to maximize operational performance.

Formulation and solution of a dynamic layout problem largely depend on whether the departments are equal or unequal sized. We focus on the equal sized departments, which represent machine tools used in production. Reviews for equal sized dynamic facility layout problem, alternative layout formulations (including different operational performance measures) and distributed layouts are given in the remaining of this chapter.

2.1 Dynamic Facility Layout Problem (DFLP) Formulation

Material flow between each pair of departments is an aggregate flow matrix (over all product types) for static facility layout problem (SFLP). This matrix can change over time because of the changes in product demand mixes, product routings and production amounts. This dynamic behavior can be modeled by adding a machine or department relocation capability to SFLP. Formulation of DFLP by McKendall et al. (2006) for a T period, N department location problem is:

$$\text{Min} \quad \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{N} A_{tijl} Y_{tijl} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} C_{tijkl} X_{tij} X_{tkl} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} X_{tij} = 1 \qquad i = 1,\ldots,N \qquad t = 1,\ldots,T \tag{2.2}$$

$$\sum_{i=1}^{N} X_{tij} = 1 \qquad j = 1,\ldots,N \qquad t = 1,\ldots,T \tag{2.3}$$

$$Y_{tijl} = X_{(t-1)ij} X_{til} \qquad i, j, l = 1,\ldots,N \qquad t = 2,\ldots,T \tag{2.4}$$

$$X_{tij} \in \{0,1\} \qquad i, j = 1,\ldots, N \qquad t = 1,\ldots, T \tag{2.5}$$

$$Y_{tijl} \in \{0,1\} \qquad i, j, l = 1,\ldots, N \qquad t = 2,\ldots, T \tag{2.6}$$

where

$$X_{tij} = \begin{cases} 1 & \text{if department } i \text{ is assigned to location } j \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{tijl} = \begin{cases} 1 & \text{if department } i \text{ is shifted from location } j \text{ to } l \text{ at the beginning of period } t \\ 0 & \text{otherwise} \end{cases}$$

A_{tijl} : cost of shifting department *i* from location *j* to *l* in period *t* (where $A_{tijj} = 0$)

C_{tijkl} : cost of material flow between department *i*, located at location *j*, and department *k*, located at *l*, in *t*

Objective function (2.1) aims to minimize total relocation cost and full material handling system (MHS) flow cost. Constraints (2.2) and (2.3) ensure that each department is assigned exactly to one location and each location is assigned exactly to one department. Constraint set (2.4) defines the relocations between consecutive periods. Finally, constraints (2.5) and (2.6) define the decision variables as binary. This is a generalized quadratic assignment problem (QAP) formulation.

Generally, A_{tijl} is defined only for departments independent of periods and locations, and C_{tijkl} is taken as the product of rectilinear distance between locations *j* and *l* and cost of the workflow per unit distance between departments *i* and *k* independent of the periods. For an *N* department *T* period problem, there are $(N!)^T$ different layout plans, which make it computationally inefficient to find the optimal plan for large problems. Hence, heuristic approaches are used to solve dynamic layout problems.

A schematic view of this problem is given in Figure 2.2 for a 9 department 3 period problem by using a discrete equal sized department representation. While passing from the first to the second period, there are no relocations. From the second to the third periods, locations of some departments (2, 4, 5, 9) change, incurring a relocation cost. For each period, there are material handling costs for each pair of departments depending on the distance and the amount of material flow between them.

Solution approaches for DFLP are explained in the following section.

Period 1		
8	3	6
1	7	4
9	2	5

Period 2		
8	3	6
1	7	4
9	2	5

Period 3		
8	3	6
1	7	9
5	4	2

Figure 2.2. Schematic view of a (9 department-3 period) problem

2.2 DFLP Solution Approaches

Dynamic Programming (DP)

Rosenblatt (1986) has the first DP study for DFLP. The recursive formulation is as follows.

$$L_{tm} = \min_k \{L_{t-1,k} + C_{km}\} + Z_{tm} \qquad t = 1,\dots,n$$

$L_{01} = 0$ assuming there is a single initial layout

where C_{km} shows the relocation cost from layout k to m, Z_{km} shows total material handling cost and L_{tm} shows minimum total cost up to period t. States are the layout configurations of each period and stages are the periods. Since "computation time usually increases exponentially with number of states in a dynamic programming problem", heuristics are developed for reducing the number of states. Rosenblatt (1986) reduces the number of states by finding the optimal static solutions independently for all periods and uses those single period solutions as the states of each period. Balakrishnan (1993) proposes a fathoming procedure for Rosenblatt's DP procedure.

Batta (1987) establishes a class of best possible upper bounds for DFLP. According to Batta's theorem, if the same layout is kept in all time periods $t = 1,\dots,T$, the problem can be solved as a static facility layout problem (SPLP) in which the from-to matrix is obtained by adding the from-to matrices in periods $t = 1,\dots,T$.

Balakrishnan et al. (1992) add a budget constraint for the relocation cost and solve this problem by two methods. First one is the DP algorithm with two state variables, one for the layouts and the other for the amount of the constrained resource available. Second algorithm is based on a singly constrained network model. Nodes are the static layouts for each period and arcs represent the cost for relocations within nodes (layout of consecutive periods). They use three different methods to determine the nodes. These are: using the best static layout of each period as nodes, using random layouts and using a combination of the first two. By relaxing the budget constraint and considering it as another objective, they model the problem as a bi-criteria shortest-path model. Solving the problem provides pareto optimal paths. They use shortest-path simplex algorithm to solve the network model.

Urban (1998) developed improved bounds for DFLP. An optimal procedure for the case of fixed arrangement costs is proposed based on incomplete dynamic programming. This procedure is also used to develop an upper bound for the general problem.

Erel et al. (2003) use a three phase approach. First phase includes finding "viable layouts" that are likely to appear in the optimal solution. Their viable layouts are the ones performing best with respect to flow data of a single period, or a combination of flow data of two or more successive periods with a weighting scheme. Moreover the set is extended by "isomorphic layouts" (layouts with the same interdepartmental distances) and screened for multiple occurrences of the same layout. Shortest path problem is solved by DP in the second phase. Final phase includes local improvement to find a number of good solutions.

Pairwise Exchange Heuristic

Computerized Relative Allocation of Facilities Technique (CRAFT) is a solution strategy for SFLP in which an initial layout is improved by pairwise exchanges. CRAFT is an improvement heuristic which cannot guarantee to find the global optimum.

Urban (1993) proposes a heuristic for DFLP using steepest descent pairwise interchange procedure (similar to that used in CRAFT) and forecast windows. First, forecast window is assigned to one and period 1 is solved with period 1 data, period 2 is solved with period 2 data, and so on. Then forecast window is increased to two. Workflow of periods 1 and 2 are combined to find the layout of period 1, workflow of periods 2 and 3 are combined to find the layout of period 2, and so on. Solutions are found by steepest descent pairwise interchange procedure and the solution of the current period is used as the initial for the next period. At the end of each iteration (forecast window), total cost of the current layout plan is calculated and finally the best one is selected.

Lacksonen and Enscore (1993) modified CRAFT to solve DFLP where pairs of locations are analysed for exchanging over all consecutive blocks of time periods.

Balakrishnan et al. (2000) show two improved pairwise exchange heuristics. First one involves working backward from the final solutions obtained by Urban's (1993) heuristic. They try pairwise exchanges from the period before the final up to the first period. Initial solutions are the ones generated by Urban's (1993) solution for each forecast window. Second heuristic is a dynamic programming approach like Rosenblatt's (1986) heuristic. Single period layout results of Urban's (1993) heuristic are used as the states of DP.

Modification of Quadratic Assignment Algorithms

Lacksonen and Enscore (1993) modify five algorithms of the static problem to include dynamic aspects. Those are CRAFT (which is explained), cutting planes, branch and bound, dynamic programming and cut trees.

Tabu Search (TS)

Kaku and Mazzola (1997) define a tabu search heuristic for DFLP. Neighborhood search is defined as interchanging two department-to-location assignments in a single period. "A move is defined to be tabu if it returns both departments to locations that they previously occupied". If the neighborhood solution yields an improved objective function value, a tabu move is permitted as an aspiration criterion. Two stopping conditions are the limits on maximum number of iterations and maximum number of consecutive moves allowed to occur without improvement on incumbent solution. Different diversification and intensification strategies are tested. Modified heuristic contains an adjustment for intensification by making changes in the size of the tabu list during the search. DFLP tabu search heuristic contains two stages. In the first stage, a diversified set of starting solutions are generated by using a construction heuristic and tabu search is applied to those initial solutions. In the second stage, best solutions are used as initial points of a more intensive search.

Genetic Algorithms (GA)

Conway and Venkataramanan (1994) use a genetic search procedure for DFLP. A solution string contains every department according to the sequence of locations for each period (30x5 genes for a 30 department 5 period problem). Two strings are selected for cross-breeding according to the "distribution of the relative strength of the strings to the entire strength of the population". Strings are split at a random splicing position. Some modifications are made to prevent infeasibilities such as having the same department twice in one period by assigning the department at the same location of previous or next period layout if possible or assigning randomly a feasible department. The string with the highest fitness value is allowed to survive into the next generation and all of the other members of the population are reproduced every generation.

Balakrishnan and Cheng (2000) developed an improved genetic algorithm. String representation is the same as that of Conway and Venkataramanan (1994). They use a nested loop GA. In the inner loop, a pair of parents is chosen randomly for a point-to-point crossover. Departments are swapped at all positions consecutively from first position of first period to last position of last period. Therefore, for an n period t department problem, $2(nt-1)$ child strings are reproduced. Mutation is applied with a small probability to minimum cost feasible child by interchanging two departments in one period. The minimum cost feasible child string replaces the maximum cost parent in the population. Inner loop ends when the difference between best child layouts in two successive generations is less than a very small threshold value. Outer loop, in which some of the poorest layouts of inner loop are replaced with random ones, continues for a predetermined number of iterations.

Chang et al. (2002) developed a symbiotic evolutionary algorithm (SymEA) for DFLP. SymEA uses a multi population idea such that each population corresponds to layouts of one period. “The unit of evaluation and selection is symbion which consists of randomly selected individuals from each population, and top ranked symbia are used as seeds to reproduce populations for the next generation”. They concentrate on the effect of symbiotic coevolution, hence they do not adopt any crossover operator. They only use mutation as a genetic operator in which two randomly selected departments are swapped.

Balakrishnan et al. (2003) propose a hybrid genetic algorithm in which they use DP for crossover. String representation is the same as that of Conway and Venkataramanan (1994). By tournament selection s strings are selected and, for a p period problem, each one is cut into p equal parts (giving single period layouts). DP in which layouts coming from a period of selected strings form the states of that period (there are s states per stage and p stages) is used for crossover. Weakest parent in the parent pool is replaced with the offspring. Mutation, in which the layout of a selected period is improved by CRAFT (including relocation cost) is applied with a rate of 5%. Process ends when a prespecified number of generations are reached. Two methods are used to generate the initial parent pool. First is random layouts and second is Urban's (1993) pairwise exchange heuristic.

Simulated Annealing (SA)

Baykasoğlu and Gindy (2001) apply a simulated annealing algorithm for DFLP. Their results are later corrected by Baykasoğlu and Gindy (2004). Their neighborhood definition is swapping (determine two different locations from the randomly selected period and swap the facilities in these locations). Initial acceptance probability is taken as P_c = 0.95. Initial temperature of the algorithm is a function of lower and upper bounds of the problem and initial acceptance probability and calculated as $T_{initial} = (f_{\min} - f_{\max}) / \ln P_c$. Length of markov chain is defined by the product of number of periods and number of departments. Final temperature is found from lower and upper bounds of the problem and the final acceptance probability, which is taken as P_f=10^{-15}. Cooling ratio (α) is calculated from a specified number of outer loop iterations (el_{max}) as $\alpha = (\ln P_c / \ln P_f)^{1/(el_{\max}-1)}$. Final temperature can also be calculated from α and el_{max} as $T_{final} = T_{initial} x \alpha^{el_{\max}}$. For T_{final}, if calculated value (from α and el_{max}) and estimated value from the final acceptance probability differs, then el_{max} and α values are modified iteratively.

Erel et al. (2003) also apply two simulated annealing approaches differing from Baykasoğlu and Gindy (2001) by the settings they use. In the first one, they use a fixed initial temperature of 5000, cooling ratio of 0.998 and maximum number of outer loop iterations of 5000. Other settings are same as in Baykasoğlu and Gindy (2001). In the second one, initial temperature and the number of outer loop iterations are found as proposed by Baykasoğlu and Gindy (2001) with cooling ratio = 0.998 and final temperature = 1. Their results are later corrected by Erel et al. (2005).

McKendall et al. (2006) use different SA settings from the previous studies. Neighborhood move definition is an iteration of random descent pairwise exchange heuristic. Initial temperature is chosen such that probability of accepting a neighboring solution with a cost of 10% above the cost of the initial solution is 0.25 ($T_{initial} = -\Delta TC / \ln(P(\Delta TC))$. Final temperature is set to 0.01. Cooling ratios and number of inner loop iterations for individual problem sizes are set experimentally. They have a second heuristic in which they add a “look ahead/back strategy” to the SA algorithm. After performing an iteration of random descent pairwise exchange by randomly selecting a period t and two departments u and v for exchange in period t, if the new (neighbor) solution is accepted either because of improvement or probabilistic selection, then the same exchange is tested for succeeding and preceding periods as long as new neighbor solutions are accepted.

Table 2.2. Comparison of DFLP solution Approaches

Author	Year	Heuristic	Basis for Comparison	Result
Rosenblatt	1986	Dynamic Programming		
Balakrishnan et al.	1992	Dynamic Programming		
Lacksonen and Enscore	1993	5 Different Heuristics (L&E)		
Urban	1993	CRAFT with forecast windows (HEUR3)		
Conway and Venkataramanan	1994	Genetic Algorithm (CONGA)		
Kaku and Mazzola	1997	TabuSearch (TS)	L&E (cutting plane) and HEUR3 on data of Lacksonen and Enscore(1993)	L&E dominates HEUR3 and TS found best for one third and matched solution quality for half of 32 problems
Urban	1998	Dynamic Programming		
Balakrishnan et al.	2000	Improved pairwise exchange (COMBINE CRAFT, BACKWARD CRAFT)	HEUR3 with initial improved (CRAFT) layouts and initial random layouts, Rosenblatt (1986)'s DP with static layouts generated with CRAFT and static layouts generated randomly	Worth implementing improvements to HEUR3
Balakrishnan and Cheng	2000	Genetic Algorithm (NLGA)	CONGA on 48 problems	NLGA generally performs better than CONGA
Baykasoğlu and Gindy*	2001	Simulated Annealing (SA_BG)	CONGA, NLGA on 48 problems of Balakrishnan and Cheng (2000)	SA_BG is the best
Chang et al.	2002	Symbiotic Evolutionary Algorithm (SymEA)	CONGA, NLGA, SA_BG on 48 problems of Balakrishnan and Cheng (2000)	SymEA outperforms CONGA and NLGA but SA_BG is the best

Table 2.2. (Continued)

Author	Year	Heuristic	Basis for Comparison	Result
Erel et al.**	2003	DP approaches (DP_10,DP_10I,DP_5,DP5I), SA approaches (SA_EG_1,SA_EG_2)	CONGA,NLGA,SA_BG on 48 problems of Balakrishnan and Cheng (2000)	SA_BG has the best for most problems but the (SA_BG) correction at 2004 shows that Erel's algorithms vastly outperform SA_BG
Balakrishnan et al.	2003	Genetic Algorithms (GADP(R), GADP(U))	CONGA, NLGA, SA_BG on 48 problems of Balakrishnan et al. (1992)	GADP (U) provides better results than using GA alone and SA_BG generally has a better average but after the SA_BG correction GADP generally outperforms SA_BG (Balakrishnan and Cheng, 2006)
Mckendall and Shang	2006	Hybrid Ant Systems (HAS I, HAS II, HAS III)	1 – L&E, TS on 30 problems of Lacksonen and Enscore (1993) ; 2- corrected SA_BG, GADP on 48 problems of Balakrishnan and Cheng (2000)	HAS III outperforms other heuristics
Baykasoğlu et al.**	2006	Ant colony optimization (ACO)	NLGA, CONGA, DP_10, DP_10I, DP_5, DP_5I, SA_EG_1, SA_EG_2 on 48 problems of Balakrishnan and Cheng (2000)	ACO performs better than NLGA and CONGA but is not as good as Erel's algorithm (2003)
Mckendall et al.**	2006	Simulated Annealing (SA I, SA II)	Best DP of Erel (2003), Best SA of Erel, Best GADP, BEST HAS on 48 problems of Balakrishnan and Cheng (2000)	SA Heuristics performed well

*Results of Baykasoğlu and Gindy (2001) are corrected in Baykasoğlu and Gindy (2004)

**Results of Erel et al. (2003) are corrected in Erel et al. (2005). Baykasoğlu et al. (2006) and Mckendall et al. (2006) used the wrong results for comparison but corrected ones are worse.

Ant Colony Optimization (ACO)

First application of ACO for DFLP is done by McKendall and Shang (2006). They proposed three hybrid ant system heuristics. Baykasoğlu et al. (2006) also applied ACO for DFLP. Their heuristic is for budget constrained and unconstrained DFLP.

Comparison of DFLP Solution Approaches

Comparison of DFLP solution approaches are given in Table 2.2. According to these results, simulated annealing algorithm with a look ahead/back strategy (McKendall et al., 2006) outperforms other algorithms in terms of the objective function value. Largest problems with 30 departments and 10 periods are solved in an average of 7.8 minutes on a Pentium IV 2.4 GHz PC. This is faster than the approaches compared by McKendall et al. (2006). Since each algorithm is tested in different computational environments (with different programming techniques), it is not possible to directly compare different heuristics for computational efficiency. Further comparisons including our results will be provided in Chapter 3.

2.3 Alternative Performance Measures in Layout Formulations

Besides the Quadratic Assignment Problem (QAP) formulations, there are alternative models which consider different metrics in the objective function, such as work-in-process inventory (WIP) level, throughput rate, cycle time and amount of empty moves of material handling system (MHS). Queuing network applications are often used in alternative formulations. Generally, these models are proposed for static facility layout problems, but we review them because we are concerned with metrics other than material flow cost.

Kouvelis and Kiran (1990) present a more comprehensive model of the plant layout problem which incorporates the throughput related aspects of automated manufacturing systems. They analyze the problem by using a closed queue network (CQN). MHS is a node in CQN and limited number of fixture pallets in an automatic manufacturing system corresponds to the finite number of jobs circulating in a CQN. Their assumptions are:

> "1) There is adequate local buffer capacity at the stations to accommodate parts.
> 2) First come first served (FCFS) queue discipline is in effect.
> 3) Service times at the stations are exponentially distributed.
> 4) The transit times are distributed according to a general distribution."

Moreover, they modify the QAP formulation by adding a WIP cost element which should satisfy the throughput requirements. They present a branch and bound procedure based on an exact procedure for their modified QAP model, which is a nonlinear integer programming formulation.

Fu and Kaku (1994) consider queuing effects in a general job shop where throughput is fixed, but WIP is not. They claim that an open queuing network is the more appropriate model. WIP can accumulate either at the machines' inbuffer or in process (L_i), machines' outbuffer waiting for transporter (L_0) or in transporter ($L_{i,j}$). They try to find the best assignment of machines to locations to minimize the WIP. Their assumptions are:

"1) External part type arrival processes into the system are Poisson.
2) Processing times at a department are i.i.d. exponential.
3) Travel times of forklifts to a department are exponential and independent of the department.
4) Buffer sizes are sufficiently large such that blocking is negligible.
5) Service discipline is first-come, first-served (FCFS)."

Moreover, they state that it is not necessary to assume exponential travel times between departments since they model these travel requests as infinite server queues. Also, multiple visits to a department are allowed in their formulation. They find that only the $L_{i,j}$ factor depends on the layout (on distances between departments). By this way they show the equivalence of total $L_{i,j}$ to the QAP objective (sum of distance-weighted material flows) meaning that under certain assumptions, QAP formulation minimizes WIP. They use a secondary measure (based on expected travel time of empty forklift to arrive at the demanding station) to discriminate the near optimal QAP solutions that minimize WIP.

Fu and Kaku (1997) state that the environment they study is different from a past study of automated guided vehicle system (AGVS) design in which AGVs move along fixed unidirectional paths. In their job shop environment, "there are multiple routes that can be traveled in either direction, blocking is not typically a concern, and demand for forklift service is not likely to be at some constant rate". Their measure for expected travel time for an empty forklift is defined as:

$$\text{Expected travel time for an empty forklift} = \sum_{i=1}^{N-1}\sum_{j=2}^{N} \frac{d_{ij}}{v}\frac{f_i^{\rightarrow}}{f}\frac{f_j^{\leftarrow}}{f} \tag{2.7}$$

where

$f_i^{\rightarrow} = \sum_{\mathrm{j}} f_{ij}$ is the total flow from department i

$f_i^{\leftarrow} = \sum_{\mathrm{j}} f_{ji}$ is the total flow into department i

$f = \sum_{i} f_i^{\rightarrow} = \sum_{i} f_i^{\leftarrow}$ is the total flow in system

v is the average velocity of a forklift

d_{ij} is the distance between locations of departments i and j

Equation 2.7 comes from the idea that forklift is at department j with probability $f_j^{\leftarrow}/f$ (since forklift stays in the last department it delivered until a request comes) and a transport request comes from department i with probability $f_i^{\rightarrow}/f$.

Azadivar and Wang (2000) optimize the facility layout by using simulation and a genetic algorithm. They use simulation to evaluate the objective functions of layout strings where the objective function is a measure of an actual system performance.

Johnson (2001) states the importance of empty vehicle traffic in AGVS design problems that concentrate on flow path design - determining travel direction of unidirectional arcs in the AGVS network - and fleet sizing. First an expression for empty trip traffic is developed for FCFS dispatching policy (requests for transport wait in a FCFS queue until a vehicle is available, and a vehicle is selected randomly or in a cyclical manner from available ones). Expected length of an empty trip between departments i and j is defined as in Fu and Kaku (1997). Additionally, they define the expected number of trips from i to j as the product of probability of an empty trip from i to j and total number of trips. This is given in Equation 2.8 using the notation of Fu and Kaku.

$$\text{Expected number of trips from } i \text{ to } j = \frac{f_i^{\leftarrow}}{f}\frac{f_j^{\rightarrow}}{f} f = \frac{f_i^{\leftarrow} f_j^{\rightarrow}}{f} \tag{2.8}$$

Next, an expression for empty trip traffic is developed for nearest vehicle rule (closest vehicle is selected from available multiple ones). Finally, they show that empty trip information can significantly improve the performance of the final AGVS design.

Castillo and Peters (2002) propose the integration of unit load and material handling considerations in facility layout design. They formulate a stochastic model that captures the operational characteristics. They try to find the best machine-to-department location assignments and unit load sizes between departments for each product type to minimize the WIP. WIP is taken as the product of demand and expected time in the system (Little's formula). In their formulation, expected time in the system of a unit load of part type q is the time spent in the processing route (sequence). For each department in the route, this time includes sum of expected waiting time of a unit load in the department, processing time of a unit load, expected waiting time of a unit load departing from the department, and travel time to the next department with pick-up/drop-off time. Their model LDP1 tries to minimize total WIP cost of each product type calculated from stochastic models with constraints for machine assignments (each machine must be assigned to one department of appropriate size), utilization levels (should be less than one for a material handling device) and unit load size levels (should be less than available material handling device capacity). They propose a simulated annealing algorithm to solve this problem. They additionally present uQAP based on QAP formulation and muQAP based on adding empty trips to the uQAP objective which decide on machine-to-department location assignments and unit load sizes. The simulated annealing algorithm is adopted to solve these modified QAP formulations. They define the approximated rate of empty trips from department *m* to department *n* as in Equation 2.9 which is similar to the work of Johnson (2001).

$$G_{mn} = \frac{(\text{ unit load arrival rate to } m)(\text{ unit load departure rate from } n)}{\text{arrival rate to the material handling system}} \tag{2.9}$$

They conclude that "it can be presumed that unit load sizes between departments have a greater impact on the expected WIP in the system than the assignment of machines to department locations. Costly structural changes to the layout design could be avoided if operational changes to the installed material handling capacity are prescribed efficiently".

Benjaafar (2002) models the congestion in the design of facility layouts in terms of WIP. The representation of one trip of the transporter is given in Figure 2.3. According to Figure 2.3, transporter moves empty from its last delivery (input buffer of station 1 where it waits until a request from station 2 arrives) to output buffer of station 2 for a pick-up. Then, it moves full from station 2 to station 4 where it drops off the load and waits for its next request. We can call this 1-2-4 trip as an *r-i-j* trip. Expected value and variance of trip time are developed from the probability of an *r-i-j* trip. Probability of a trip from *i* to *j* is defined as the ratio of total amount of workload between departments *i* and *j* to total amount of workload of transporter. This definition is also valid for a multiple vehicle system. Benjaafar defines a closed form expression of transporter utilization as the sum of full (loaded) and empty (unloaded) utilizations. Empty utilization is defined as in Equation 2.10. Transporter moves from department *r* to department *i* according to the probability that transporter was located at *r* (λ_r / λ_t) and the workload of *i* (λ_i). The idea for empty trips is similar to that in

Castillo and Peters (2002).

$$\text{Empty Utilization} = \sum_{r=1}^{M+1}\sum_{i=0}^{M}(\lambda_r \lambda_i / \lambda_t) t_{ri}(x) \tag{2.10}$$

where

λ_i : total amount of workload of department i

λ_t : total amount of workload of transporter

$t_{ri}(x)$: traveling time of transporter from r to i for a layout configuration x

M : number of departments

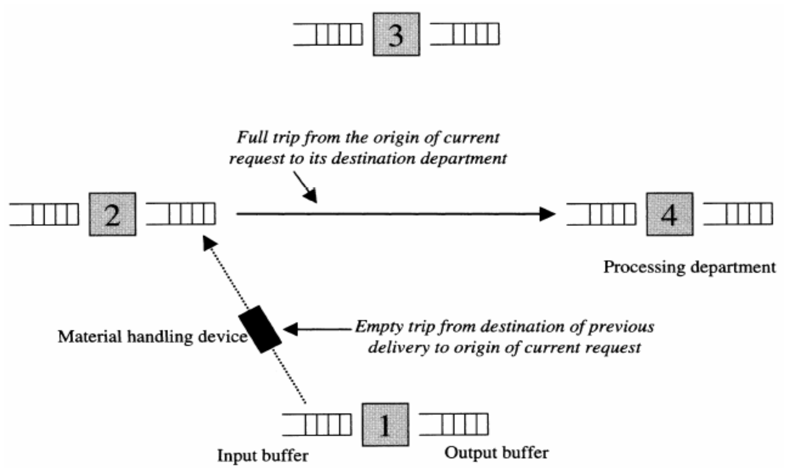


Figure 2.3. An empty trip followed by a full trip of transporter (Benjaafar, 2002)

Closed form WIP approximation is dependent on the squared coefficients of variation of job interarrival times and processing times of departments. These are calculated by using the utilization of departments and full and empty utilization of the material handling system. By the assumptions of Fu and Kaku (1994), WIP formulation is equivalent to the QAP objective. Benjaafar's (2002) model aims at minimizing WIP at each station and the transporter by assigning each department to a location and satisfying the transporter utilization constraint. For the multiple transporter case, Benjaafar (2002) shows that the random dispatching rule gives the same probability of an *r-i-j* trip as in the single transporter case. According to Benjaafar's experiments, QAP formulation can yield infeasible layouts in terms of material handling utilization and also does not minimize WIP. Moreover, WIP is affected by both full and empty utilizations of the transporter, but minimizing total (full plus empty) utilization of transporters does not always lead to a smaller expected WIP. He states that "this may cause an increase in squared coefficient of variation of expected *r-i-j* trip time which could be sufficient to either increase material handling WIP or increase the arrival variability at the processing departments, which in turn could increase their WIP".

Curry et al. (2003) state that Johnson (2001) and Benjaafar (2002) use state independent service time approximations and "the state-dependent nature of service times leads to inaccuracies in the general distribution model approximations that are based on the standard Poisson model paradigm". They use nearest vehicle rule for transporter requests and model the empty travel time of the transporter. They conclude that "the approach is computationally tractable for small numbers of transporters, but the computational burden of the approach grows exponentially with the number of transporters supporting the system".

In the above studies, empty trips of material handling system are usually taken into account in the same way. It is generally seen that queuing network approaches (under certain approximations) are used to estimate the performance measures such as WIP, flow time, full and empty utilization of transporters. A generic model can have multiple objectives with combination of these measures.

The relationship between the total travel cost of full transporters and WIP is investigated in the literature. It is shown by Benjaafar (2002) that minimizing this cost also minimizes WIP only under certain assumptions. In the special case where empty travel time is negligible and all interarrival, processing and transportation times are assumed to be exponentially distributed, Benjaafar's (2002) closed form expression for the expected WIP accumulated by transporter, $E(\mathrm{WIP}_t)$, becomes the only WIP factor that is a function of the layout and $E(\mathrm{WIP}_t)$ is minimized if full transportation cost is minimized.

All the models discussed in this section are restricted to single period layout design problems and have not yet been extended for dynamic facility layout design which has an additional cost element of machine relocations.

2.4 Distributed Layouts

Having alternative resources can provide an improvement in operational performance.

“Distributed layouts disaggregate large functional departments into subdepartments distributed throughout the plant. Such layouts are especially appealing when demand fluctuates too frequently to make reconfiguring the plant cost effective” (Benjaafar et al., 2002). This can possibly decrease the transportation costs since it relaxes the idea of grouping the same type of machines in neighboring locations as in the conventional process layout. In the distributed layout problem, besides the machine-to-location assignments, processes should also be assigned to one of the duplicate machines considering the capacity restrictions.

There are models that integrate location decisions and flow allocation among duplicate machines, for example, Urban et al. (2000), Castillo and Peters (2003). These approaches are generally limited with single period design.

Benjaafar and Sheikhzadeh (2000) integrate the location and flow allocation decisions in an uncertain production environment where duplicates of departments exist. They try to find the best layout and flow allocation for a set of demand scenarios with known probabilities (which are integrated into their model). Their objective is to minimize the expected material handling cost. They call the solution of the model “distributed layout”. They test this layout with different levels of duplication and find that it has the minimum objective compared to functional, maximally distributed (duplicates are uniformly distributed throughout the floor) and random layouts. Illustrations of those layouts are given in Figure 2.4.

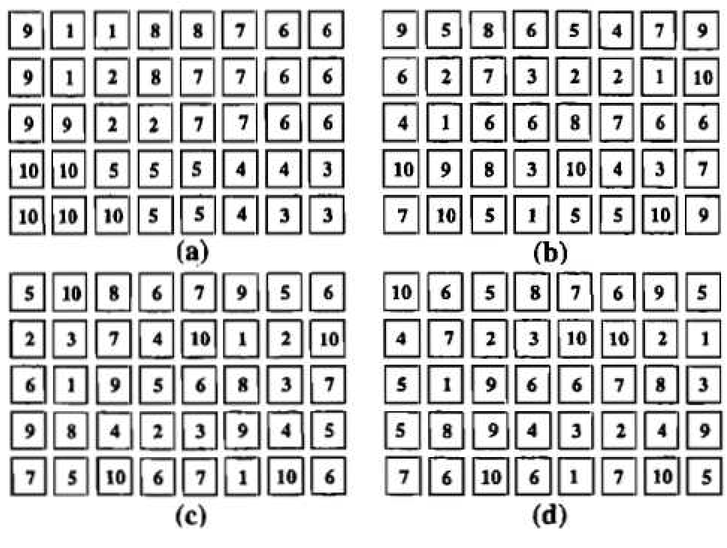


Figure 2.4. a) functional, b) distributed, c) maximally distributed, d) random layout representations (Benjaafar and Sheikzadeh, 2000)

They also observe that "increased duplication is always better. Effect of duplication is of the diminishing kind, with most of the benefits realized with the initial disaggregation of departments into two subdepartments. Further disaggregation, for all the observed cases, yields only marginal improvements". It should be stated that those improvements are in terms of the expected total full trips of MHS. This is a single period problem but includes different demand scenarios within the model.

To the best of our knowledge, the only multi-period study in this area is that of Lahmar and Benjaafar (2005). Although distributed layouts are used when relocation cost and demand uncertainty is high (Benjaafar et al., 2002), Lahmar and Benjaafar (2005) use the idea to respond to known future changes in demand by allowing both distribution and relocation. They modify DFLP by adding machine or department duplicates to the model. Besides the machine locations, they make the flow allocation decision for duplicate machines in the multi-period problem. Their formulation is as follows.

$$Min\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{n=1}^{N_i}\sum_{j=1}^{N}\sum_{m=1}^{N_j}\sum_{k=1}^{M}\sum_{l=1}^{M} v_{nimjpt}x_{nikt}x_{mjlt}c_{kl}d_{kl} + \sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{n=1}^{N_i}\sum_{k=1}^{M}\sum_{l=1}^{M} x_{nikt}x_{nil(t+1)}r_{nikl} \quad (2.11)$$

s.t

$$\sum_{i=1}^{N}\sum_{n=1}^{N_i} x_{nikt} = 1 \qquad \forall k,t \quad (2.12)$$

$$\sum_{k=1}^{M} x_{nikt} = 1 \qquad \forall i,n,t \quad (2.13)$$

$$\sum_{m=1}^{N_j}\sum_{n=1}^{N_i} v_{nimjpt} = v_{ijpt} \qquad \forall i,j,p,t \quad (2.14)$$

$$\sum_{i=0}^{N}\sum_{n=1}^{N_i} v_{nimjpt} = \sum_{q=0}^{N}\sum_{r=1}^{N_q} v_{mjrqpt} \qquad \forall j,m,p,t \quad (2.15)$$

$$\sum_{p=1}^{P}\sum_{i=0}^{N}\sum_{n=1}^{N_i} v_{nimjpt}t_{mjp} \le C_{mj} \qquad \forall j,m,t \quad (2.16)$$

$$x_{nikt} \in \{0,1\} \qquad \forall i,n,k,t \quad (2.17)$$

$$v_{nimjpt} \ge 0 \qquad \forall i,n,j,m,p,t \quad (2.18)$$

where

$$x_{nikt} : \begin{cases} 1 & \text{if } n^{th} \text{ duplicate of department type } i \text{ is assigned location } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

v_{nimjpt} : volume of flow due to product p between n^{th} duplicate of department i and m^{th} duplicate of department j in period t

v_{ijpt} : total flow volume due to product p between department duplicates of type i and department duplicates of type j in period t

d_{kl} : travel distance between location k and location l

t_{nip} : processing time per unit load of product type p at department duplicate n of type i

c_{kl} : cost (per unit distance) of moving a unit load from a department located at k to a department located at l

C_{ni} : capacity (available operation time) of duplicate n of department type i

r_{nikl} : cost of rearranging duplicate n of department type i from location k to l

T : total number of periods

N_i : total number of department duplicates of type i

N : total number of department types

M : total number of locations

P : total number of product types

Objective function (2.11) includes full trips of MHS and relocation costs. Constraints (2.12) and (2.13) ensure one-to-one assignment of department duplicates and locations. Constraints (2.14) equate the flow between duplicates of type i and type j to total flow between types i and j. Constraints (2.15) ensure flow balances. Constraints (2.16) ensure that the workload (in terms of time) of each department duplicate does not exceed its capacity. Constraints (2.17) define the location variables as binary and constraints (2.18) ensure all flows are positive.

If the locations of department duplicates in each period are known, this quadratic problem decomposes into P linear programming (LP) problems. They propose two search algorithms, A1 and A2, to solve this problem. In the first iteration of A1, they improve an initial layout (having its optimal flow allocation matrix) by a modified 2-opt heuristic. Then, they determine the new flow allocation according to the final location decisions by solving P LPs. They continue in this manner while there is improvement in total cost compared to the cost of previous iteration. A2 differs from A1 in that they compute the corresponding optimal flow allocation for every pairwise interchange carried out in the modified 2-opt heuristic. They experimentally investigate the effects of duplication level, relocation cost, flow variability and product variability.

As a result, for distributed layouts, only the full trips of MHS (additionally relocation cost for dynamic case) are taken into account, and effects of duplication are investigated in the literature.

2.5 Conclusion

According to this review, SA is a promising solution method for DFLP. Some modifications on SA can improve its performance. SA can also be adapted to solve DDFLP.

There are studies trying to add different performance metrics (other than cost of full trips of MHS) into the layout models. However, those studies are limited to single period problems.

Dynamic layout model is improved by modifying it to include the distributed layout decisions. Therefore promising methodologies of DFLP can be implemented to solve the dynamic distributed case. In this case, flow allocations must also be decided in addition to the locations. Effects of distribution levels are investigated only on relocation and full transportation costs. However, their effects on other operational performance metrics such as WIP and machine utilizations are unknown.

According to these conclusions, we first propose a new algorithm to solve DFLP based on an existing SA algorithm and show that it is a promising method. We then modify our new algorithm to solve the dynamic distributed facility layout problem which tries to minimize total transportation (full and empty trips of MHS) and relocation costs. Finally, we use simulation to explore operational performance of varying duplication levels and demand characteristics for the layouts obtained by our SA algorithm.

CHAPTER 3

A NEW SIMULATED ANNEALING ALGORITHM FOR THE DYNAMIC FACILITY LAYOUT PROBLEM

According to the review in Chapter 2, SA algorithms of McKendall et al. (2006) are promising for DFLP. The first algorithm of McKendall et al. (2006), SAI, is summarized below.

Step 0: Read the flow matrices for each period, distance matrix, and rearrangement costs as input data. Define the SA parameters: T_0 the initial temperature, α the cooling ratio, A the attempted number of moves at each temperature level, and T_{min} the minimum allowable temperature.

Step 1: Initialize the temperature change counter, $r = 1$.

Step 2: (a) Generate an initial solution y^0 and assign it to the current solution, $y = y^0$.

(b) Obtain the cost of the current solution, $f(y)$.

(c) Record the best solution and best cost, Best_sol = y and Best_cost = $f(y)$.

Step 3: Initialize counter for the number of attempted moves at each temperature, $i = 0$. Set the current temperature according to the annealing schedule, $T_c = T_0\, \alpha^{r-1}$. If $T_c < T_{min}$, then explore the entire neighborhood of the Best_sol (i.e., use the steepest descent pairwise exchange heuristic), and return Best_sol and Best_cost.

Step 4: (a) Perform an iteration of the random descent pairwise exchange heuristic. In other words, randomly select a period t, and then randomly select two departments u and v in period t. Exchange the locations of departments u and v in period t, and denote the neighboring solution as y'. Also, update $i = i + 1$.

(b) Calculate the change in total cost, $\Delta TC = f(y') - f(y)$.

Step 5: If ($\Delta TC < 0$) or ($\Delta TC > 0$ and $x = random(0, 1) < P(\Delta TC) = exp(-\Delta TC / T_c)$), then set $y = y'$. If Best_cost > $f(y)$, then Best_cost = $f(y)$ and Best_sol = y.

Step 6: If $i = A$, then update $r = r + 1$, and go to *Step 3*. Otherwise, go to *Step 4*.

According to Dowsland (1995), SA differs from the random descent method since uphill moves are allowed. These moves are made in a controlled manner in SA. The randomly selected neighbor is accepted if it makes an improvement. It may also be accepted randomly

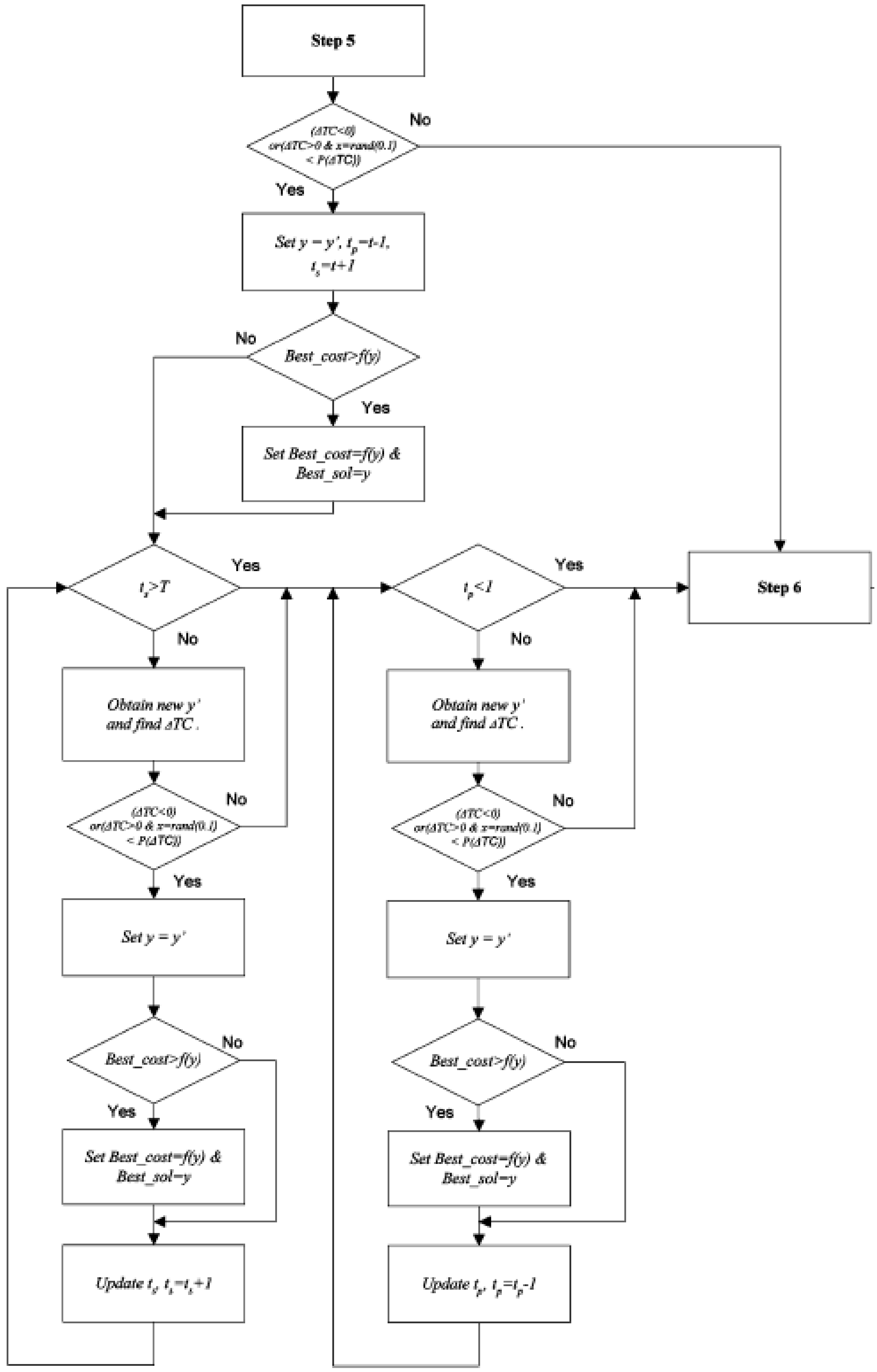


Figure 3.1. Look ahead/back strategy of SAII Algorithm, McKendall et al. (2006)

even if it gives rise to an increase in the cost function. This acceptance depends on the control parameter (temperature) and the magnitude of the increase (as in *Step 5*). The temperature is decreased at every *A* iterations (as in *Step 6* ad *3*), making acceptance of nonimproving moves less likely.

In the second algorithm, SAII, a look ahead/back strategy is added to the first algorithm. If an iteration of the random descent pairwise exchange heuristic in *Step 4* is accepted in *Step 5* of SAI, either by improvement or stochastically, then the same exchange of locations of departments *u* and *v* is tested in succeeding and preceding periods. Detailed flow chart of this process in *Step 5* is given in Figure 3.1. Settings of SAI and SAII are explained in Chapter 2.

3.1 Proposed SA Algorithm (SALAB)

Proposed algorithm, named SALAB, differs from SAII algorithm of Mckendall et al. (2006) by its neighborhood definition. Details of the algorithm are given in following subsections.

Neighborhood Definition

Neighborhood move of SALAB is defined as moving a department *n* to location *l* in period *t* where *n*, *l* and *t* are selected at random. In this move, the department originally located at *l* in period *t* is also moved to the original location of department *n*.

> $L(n, t)$: Location of department *n* in period *t*
> $D(l, t)$: Department originally at location *l* in period *t*
> $Move(n, l, t)$: Move department *n* to location *l* in period *t*
> $NeigborhoodMove(n, l, t) = Move\ (D(l, t), L(n, t), t) + Move(n, l, t)$

Neighborhood move definition of SAI (randomly select a period *t*, randomly select two departments *u* and *v* in period *t* and exchange the locations of departments *u* and *v* in period *t*) and $NeigborhoodMove(n,l,t)$ are exactly the same. However, when the look ahead/back strategy is considered, SAII tries to exchange the locations of a selected department pair in succeeding and preceding periods, whereas $NeigborhoodMove(n,l,t)$ tries to fix the location of department *n* as *l* in these periods.

Motivation for using $NeigborhoodMove(n,l,t)$ is the relocation cost factor. Relocation cost can be reduced by fixing locations of departments in consecutive periods as long as the move is acceptable. In SAII, when the algorithm starts from a layout plan where departments are randomly located, it is difficult to place a department at the same location in consecutive periods in look ahead/back iterations. However, this would be easy for an SA algorithm having a look ahead/look back strategy with the neighborhood definition *NeigborhoodMove*(n,l,t).

```
START
Initialization:
        select initial solution S0, initial temperature t0, a temperature reduction function T, inner loop
iteration limit A.
        set current temperature t= t0, out_loop_count = 0, bestsol = S0, bestcost =f(S0).
Outer loop: repeat while stopping condition is not true
        increase out_loop_count by 1
        iteration_count = 0;
        Inner loop: repeat while iteration_count is less than A
                increase iteration_count by 1
                randomly select department n, location l and period p
                execute NeighborhoodMove(n,l,p) to generate solution S from neighborhood N(S0)
                calculate change in total cost Δ = f(S) – f(S0)
                if Δ < 0
                        S0 = S
                        if f(S0) < bestcost  then bestcost= f(S0) and bestsol = S0
                        execute LAB *
                else
                        generate r uniformly distributed in the range [0,1)
                        if r < e^(-Δ/t) then  S0 = S and execute LAB *
                endif
        endwhile
        set current temperature level t = T (t0)
endwhile
END
* LAB: for department n, location l and period p
        START
        period_count = p+1
        repeat while period_count is less than or equal to the last period
                increase iteration_count by 1
                execute NeigborhoodMove(n,l,period_count) to obtain the new layout plan S
                calculate Δ = f(S) – f(S0)
                if Δ < 0
                        S0 = S
                        if f(S0) < bestcost  then bestcost = f(S0) and bestsol = S0
                else
                        generate r uniformly distributed in the range [0,1)
                        if r < e^(-Δ/t) then  S0 = S  else go to NEXT
                endif
                increase period_count by 1
        endwhile
NEXT:   period_count = p-1
        repeat while period_count is less than or equal to the first period
                increase iteration_count by 1
                execute NeigborhoodMove(n,l,period_count) to obtain the new layout plan S
                calculate Δ = f(S) – f(S0)
                if Δ < 0
                        S0 = S
                        if f(S0) < bestcost  then bestcost = f(S0) and bestsol = S0
                else
                        generate r uniformly distributed in the range [0,1)
                        if r < e^(-Δ/t) then  S0 = S else go to END
                endif
                decrease period_count by 1
        endwhile
        END
```

Figure 3.2. SALAB Algorithm

SAII is modified in this manner and named as SALAB in this study. SALAB algorithm, based on Dowsland (1995) and McKendall et al. (2006), is given in Figure 3.2.

3.2 Fine Tuning SALAB

General decisions of SA are initial temperature (t_0), cooling ratio (α), attempted number of moves at each temperature level (iteration limit, A) and stopping condition (final temperature, t_{final}). Cooling schedule is selected as $T = t_0\, \alpha^{n-1}$ where n is the outer loop count.

Fine tuning is carried out for six types of DFLP problems (Balakrishnan and Cheng, 2000) varying in size (number of departments and periods):

1. 6 Departments, 5 Periods (6D5P)
2. 6 Departments, 10 Periods (6D10P)
3. 15 Departments, 5 Periods (15D5P)
4. 15 Departments, 10 Periods (15D10P)
5. 30 Departments, 5 Periods (30D5P)
6. 30 Departments, 10 Periods (30D10P)

There are eight problems in each category. First, initial temperature and stopping conditions for which SALAB converges after making a sufficiently large number of iterations are decided on. After fixing initial temperature and stopping condition, effect of various iteration limit (A) and cooling ratio (α) combinations are searched for each of the six DFLP problem categories.

Initial Temperature and Stopping Condition

Initial temperature of SALAB is selected such that the probability of accepting a neighboring solution whose cost is 10% above the cost of the initial solution is $P(\Delta) = 0.25$. McKendall et al. (2006) also use this approach for all DFLP categories as follows.

$$P(\Delta) = e^{-\Delta / t_0} = 0.25, \quad \Delta = 0.1 f\,(initial\ solution), \quad t_0 = -0.1 f\,(initial solution)\,/\ln(0.25)$$

They use t_{final}=0.01 or a maximum CPU time for stopping condition. In SALAB, the total number of neighbor solutions visited ($A\times$out_loop_count) is used as the stopping condition. Stopping condition for each of the six problem categories is set experimentally (given in TERMCOND column of Table 3.1).

Iteration Limit and Cooling Ratio

There are several A and α combinations that are reported for SAI and SAII algorithms by McKendall et al. (2006). After setting initial temperature and stopping condition as defined above, best combination of A and α for SALAB (starting from random initial solutions) is experimentally searched for six problem categories. Results for tested configurations are given in APPENDIX A. Selected A and α settings are given in Table 3.1 for different problem categories..

Table 3.1. Selected iteration limit, cooling ratio and terminating condition settings

D	P	α	A	TERMCOND
6	5	0.997	720	3,000,000
6	10	0.997	1,440	5,000,000
15	5	0.998	2,000	10,000,000
15	10	0.998	7,500	30,000,000
30	5	0.998	10,000	50,000,000
30	10	0.998	12,000	60,000,000

SALAB algorithm is tested on 48 benchmark problems (six categories$\times$eight problems) presented by Balakrishnan and Cheng (2000). Three different initial solution strategies are tried for SALAB. SALAB-R starts with a randomly generated initial solution. SALAB-1 is initialized with the best static layout that does not change over periods. SALAB-2 starts with a layout plan in which each period is solved independently without relocation costs. In addition to these three versions of SALAB, SA*, a basic algorithm without the look ahead/back strategy, is also run for comparison purposes. Note that SA* is the same as SA I of McKendall et al. (2006). To find the initial solution for SALAB-1, an aggregate flow matrix is obtained by summing inter-departmental flows over all periods. The static initial layout is found by running SA* on this matrix where inner loop iteration limit is taken as A (from Table 3.1) divided by the number of periods, and termination condition is taken as TERMCOND (from Table 3.1) divided by the number of periods. The initial solution for SALAB-2 is found by solving each period separately as a static problem with the same settings of static initial layout of SALAB-1. All versions of SALAB are coded in C. Runs are made on a Pentium IV 1.8 GHz PC.

3.3 Experimental Results for DFLP

Detailed experimental results of the above strategies (including average deviations from the best known solutions before the runs of this study) are given in APPENDIX B. The solution quality and solution time results with improvement and convergence properties of each algorithm are given in Table 3.2.

Table 3.2. Average results for SALAB algorithms and SA*

	D	P	ADEV%	DEV%	IMP%	BEST	ACCEPT	BETTER	TERMCOND	CPU	PCPU
SA*	6	5	0.00	0.00	20.18	666,508	1,181,417	1,181,493	3,000,000	0.20	-
	6	10	0.03	0.01	19.97	1,855,341	3,031,917	3,032,035	5,000,000	0.38	-
	15	5	0.23	0.04	18.93	5,229,388	7,654,016	7,654,651	10,000,000	0.96	-
	15	10	0.23	0.07	19.44	22,644,790	29,671,653	29,672,378	30,000,000	2.96	-
	30	5	0.25	0.05	20.82	30,164,754	41,448,500	41,450,510	50,000,000	7.63	-
	30	10	0.71	0.19	20.88	40,633,841	55,571,556	55,574,609	60,000,000	9.15	-
SALAB -1	6	5		0.00	2.04	241,093	1,107,054	1,107,091	3,000,000	0.15	0.04
	6	10		0.00	3.11	1,468,221	2,807,098	2,807,145	5,000,000	0.27	0.03
	15	5		0.06	3.46	4,861,984	7,400,466	7,400,633	10,000,000	0.80	0.18
	15	10		0.20	6.65	21,341,651	29,001,637	29,002,123	30,000,000	2.26	0.27
	30	5		0.06	1.86	27,629,340	40,269,291	40,270,903	50,000,000	6.47	1.37
	30	10		0.00	3.81	38,854,531	54,930,169	54,932,603	60,000,000	7.67	0.82
SALAB - 2	6	5		0.00	7.30	258,749	1,148,700	1,148,720	3,000,000	0.15	0.16
	6	10		0.01	7.79	1,541,599	2,775,548	2,775,627	5,000,000	0.27	0.29
	15	5		0.07	3.02	4,932,749	6,792,482	6,792,832	10,000,000	0.80	0.83
	15	10		0.11	3.46	20,894,888	29,953,176	29,955,101	30,000,000	2.25	2.59
	30	5		0.06	7.31	28,529,178	39,906,381	39,910,088	50,000,000	6.46	6.58
	30	10		-0.02	7.90	39,512,908	54,892,814	54,894,409	60,000,000	7.66	8.01
SALAB - R	6	5	0.00	0.00	20.18	256,778	1,151,712	1,151,781	3,000,000	0.15	-
	6	10	0.00	0.00	19.99	1,619,148	3,051,911	3,051,985	5,000,000	0.26	-
	15	5	0.09	0.00	19.05	5,030,819	7,441,973	7,442,204	10,000,000	0.79	-
	15	10	0.17	0.05	19.49	21,916,916	29,671,912	29,672,494	30,000,000	2.23	-
	30	5	0.10	-0.10	20.94	29,023,156	41,905,960	41,907,101	50,000,000	6.44	-
	30	10	0.48	-0.04	21.06	40,096,456	55,778,508	55,781,169	60,000,000	7.64	-

ADEV% : Average percentage deviaton over 40 runs (eight problems x five runs each starting with a different random initial solution) from the best known solution = 100 x (bestcost - bestknown) / bestknown.
DEV%: For SA* and SALAB-R, average percentage deviation over the best of five runs of eight problems from the best known solution; for SALAB-1 and SALAB-2, average percentage deviation over the single run of eight problems = 100 x (bestcost - bestknown) / bestknown.
IMP% : % improvement of best cost compared to the initial cost = -100 x (bestcost - initialcost) / initialcost.
BEST : Number of moves until the best solution is found.
ACCEPT: Number of moves until the last probabilistic acceptance of a non-improving solution.
BETTER : Number of moves until the last acceptance of an improving solution.
TERMCOND : total number of neighbor solutions visited (stopping condition).
CPU : Cpu time in minutes on a Pentium IV 1.8GHz PC
PCPU: Preprocessing CPU time in minutes to find initial solution.

According to Table 3.2, random initial solutions are approximately improved by 20% with SA* and SALAB-R. The improvement is less than 8% for SALAB-1 and SALAB-2. Number of moves until best solution or until last acceptance shows the suitability of our terminating conditions.

For SA*, each problem is solved five times each starting with a different random initial solution. Average deviation over 48 problems for the best of five runs from the best known solution is 0.06%. Average deviation of all runs is 0.24%. For SALAB-R, each problem is also solved five times. Average deviation over 48 problems for the best of five runs is -0.01%. Average deviation of all runs is 0.14%. SALAB-1 and SALAB-2 are special cases of SALAB with extreme layout plans as initial solutions. Average deviation from the best known solution is 0.05% for SALAB-1 and 0.04% for SALAB-2. We have small or even negative deviations since best known solutions are found or improved several times. According to best of five runs, SA* found the best known solution for 17 problems and a better result than the best known for 9 problems, SALAB-R found the best known for 20 problems and a better solution than the best known for 15 problems. According to single runs, SALAB-1 found the best known for 17 problems and improved the best known for 10 problems; SALAB-2 found the best known for 18 problems and improved the best known for 8 problems. The small deviations from the best known results show that even without a look ahead/back strategy, the basic SA* with proposed settings in Table 3.1 is a promising algorithm. However, the algorithm can find better results with the look/ahead back strategy. Additionally, the small differences between deviations of SALAB-1, SALAB-2 and SALAB-R and the small differences between average and best results of SALAB-R show that SALAB is not dependent on the initial solution; hence a random initial solution can be used for SALAB.

In terms of computation time, SALAB is slightly faster than SA*. For the largest problem, SALAB takes 7.67 minutes per run compared to 9.15 minutes of SA*, perhaps since the neighbor solution is known in the look/ahead back procedure but it is always randomly generated in SA*. Average deviation according to best results of five runs of SALAB-R is -0.01%. This is better than 0.05% of SALAB-1 and 0.04% of SALAB-2 but achieved by running the algorithm five times with different initial solutions whereas SALAB-1 and SALAB-2, with a preprocessing time not much longer than one SALAB run, are run only once for each problem. It approximately takes 40 minutes to run SALAB-R five times for the largest problem on a Pentium IV 1.8 GHz PC. Since DFLP is not a frequently solved problem, time required for five runs of SALAB-R is affordable even for the 30D10P case.

Table 3.3. Comparison of algorithms I, results for 48 benchmark problems of Balakrishnan and Cheng (2000).

Pr #	D	P	SA I	SA II	SA_EG	SA_BG	ACO	HAS	DP	GADP	BEST
1	6	5	**106,419**	**106,419**	**106,419**	107,249	**106,419**	**106,419**	**106,419**	**106,419**	106,419
2	6	5	**104,834**	**104,834**	**104,834**	105,170	**104,834**	**104,834**	**104,834**	**104,834**	104,834
3	6	5	**104,320**	**104,320**	**104,320**	104,800	**104,320**	**104,320**	**104,320**	104,529	104,320
4	6	5	**106,399**	**106,399**	**106,399**	106,515	106,509	**106,399**	106,509	106,583	106,399
5	6	5	**105,628**	**105,628**	**105,628**	106,282	**105,628**	**105,628**	**105,628**	**105,628**	105,628
6	6	5	**103,985**	**103,985**	**103,985**	**103,985**	104,053	**103,985**	**103,985**	104,315	103,985
7	6	5	**106,439**	**106,439**	**106,439**	106,447	**106,439**	**106,439**	106,447	106,447	106,439
8	6	5	**103,771**	**103,771**	**103,771**	**103,771**	**103,771**	**103,771**	**103,771**	**103,771**	103,771
9	6	10	**214,313**	**214,313**	**214,313**	215,200	217,251	**214,313**	**214,313**	**214,313**	214,313
10	6	10	**212,134**	**212,134**	**212,134**	214,713	216,055	**212,134**	**212,134**	**212,134**	212,134
11	6	10	**207,987**	**207,987**	**207,987**	208,351	208,185	**207,987**	**207,987**	**207,987**	207,987
12	6	10	**212,530**	212,741	212,747	213,331	212,951	**212,530**	212,741	212,741	212,530
13	6	10	**210,906**	**210,906**	211,072	213,812	211,076	**210,906**	211,022	210,944	210,906
14	6	10	**209,932**	**209,932**	**209,932**	211,213	210,277	**209,932**	**209,932**	210,000	209,932
15	6	10	**214,252**	**214,252**	214,438	215,630	215,504	**214,252**	**214,252**	215,452	214,252
16	6	10	**212,588**	**212,588**	**212,588**	214,513	214,621	**212,588**	**212,588**	**212,588**	212,588
17	15	5	**480,453**	480,496	481,738	501,447	501,447	**480,453**		484,090	480,453
18	15	5	**484,761**	**484,761**	485,167	506,236	506,236	**484,761**		485,352	484,761
19	15	5	489,058	**488,748**		512,886	512,886	**488,748**	491,310	489,898	488,748
20	15	5	**484,405**	484,414	485,862	504,956	504,956	484,446		484,625	484,405
21	15	5	487,882	487,911	489,304	509,636	509,636	**487,722**		489,885	487,722
22	15	5	487,162	487,147	488,452	508,215	508,215	486,685		488,640	486,493
23	15	5	487,232	**486,779**	487,576	508,848	508,848	486,853		489,378	486,779
24	15	5	491,034	**490,812**	493,030	512,320	512,320	491,016		500,779	490,812
25	15	10	980,087	979,468		1,017,741	1,017,741	980,351	983,070	987,887	978,848
26	15	10	979,369	**978,065**	982,714	1,016,567	1,016,567	978,271		980,638	978,065
27	15	10	983,912	982,396	988,465	1,021,075	1,021,075	**978,027**		985,886	978,027
28	15	10	974,416	972,797		1,007,713	1,007,713	974,694	976,456	976,025	971,740
29	15	10	977,188	978,067	982,191	1,010,822	1,010,822	979,196		982,778	976,811
30	15	10	970,633	**967,617**	973,199	1,007,210	1,007,210	971,548		973,912	967,617
31	15	10	979,198	979,114		1,013,315	1,013,315	980,752	982,790	982,872	978,946
32	15	10	984,927	983,672	988,304	1,019,092	1,019,092	985,707		987,789	983,486
33	30	5	576,039	576,741	579,570	604,408	604,408	576,886		578,689	575,028
34	30	5	568,316	568,095		604,370	604,370	570,349	569,482	572,232	568,057
35	30	5	573,739	574,036		603,867	603,867	576,053	578,506	578,527	572,478
36	30	5	567,911	**566,248**		596,901	596,901	566,777	569,723	572,057	566,248
37	30	5	559,277	558,460		591,988	591,988	558,353	558,792	559,777	556,257
38	30	5	566,077	566,597		599,862	599,862	566,792	568,047	566,792	565,004
39	30	5	**567,131**	568,204		600,670	600,670	**567,131**	568,721	567,873	567,131
40	30	5	576,014	573,755		610,474	610,474	575,280	574,813	575,720	573,193
41	30	10	1,164,359	1,163,222	1,173,483	1,223,124	1,223,124	1,166,164		1,169,474	1,158,684
42	30	10	1,162,665	1,161,521		1,231,151	1,231,151	1,168,878	1,170,092	1,168,878	1,157,547
43	30	10	1,157,693	1,156,918		1,230,520	1,230,520	1,166,366	1,168,720	1,166,366	1,153,389
44	30	10	1,149,048	1,145,918		1,200,613	1,200,613	1,148,202	1,150,265	1,154,192	1,143,000
45	30	10	1,126,432	1,127,136		1,210,892	1,210,892	1,128,855	1,128,013	1,133,561	1,122,196
46	30	10	1,145,445	1,145,146		1,221,356	1,239,255	**1,141,344**	1,145,858	1,145,000	1,141,344
47	30	10	1,148,083	**1,140,744**		1,212,273	1,248,309	1,140,773	1,143,144	1,145,927	1,140,744
48	30	10	1,166,672	1,161,437		1,231,408	1,231,408	1,166,157	1,165,994	1,168,657	1,160,420
# of BEST			20	23	13	2	6	23	12	8	
AVG DEV %			0.16	0.10	0.25	3.60	3.65	0.18	0.30	0.46	

#ofBEST : Number of times best is found
AVG DEV%: Average deviation percentage from best known solution
SA I / SA II : McKendall et al. (2006) ; SA_EG / DP : Erel et al. (2005); SA_BG : Baykasoğlu and Gindy (2004)
ACO : Baykasoğlu et al. (2006); HAS : McKendall and Shang (2006); GADP : Balakrishnan et al. (2003)

Table 3.4. Comparison of algorithms II, results for 48 benchmark problems of Balakrishnan and Cheng (2000).

Pr #	D	P	SymEA	NLGA	CONGA	SA*	SALAB-R	SALAB-1	SALAB-2	BEST
1	6	5	**106,419**	**106,419**	108,976	**106,419**	**106,419**	**106,419**	**106,419**	106,419
2	6	5	**104,834**	**104,834**	105,170	**104,834**	**104,834**	**104,834**	**104,834**	104,834
3	6	5	**104,320**	**104,320**	104,520	**104,320**	**104,320**	**104,320**	**104,320**	104,320
4	6	5	106,509	106,515	106,719	**106,399**	**106,399**	**106,399**	**106,399**	106,399
5	6	5	**105,628**	**105,628**	**105,628**	**105,628**	**105,628**	**105,628**	**105,628**	105,628
6	6	5	**103,985**	104,053	105,606	**103,985**	**103,985**	**103,985**	**103,985**	103,985
7	6	5	**106,439**	106,978	**106,439**	**106,439**	**106,439**	**106,439**	**106,439**	106,439
8	6	5	**103,771**	**103,771**	104,485	**103,771**	**103,771**	**103,771**	**103,771**	103,771
9	6	10	214,921	214,397	218,407	**214,313**	**214,313**	**214,313**	**214,313**	214,313
10	6	10	212,138	212,138	215,623	**212,134**	**212,134**	**212,134**	**212,134**	212,134
11	6	10	**207,987**	208,453	211,028	**207,987**	**207,987**	**207,987**	**207,987**	207,987
12	6	10	212,741	212,953	217,493	212,741	**212,530**	**212,530**	212,741	212,530
13	6	10	211,072	211,575	215,363	**210,906**	**210,906**	**210,906**	**210,906**	210,906
14	6	10	**209,932**	210,801	215,564	**209,932**	**209,932**	**209,932**	**209,932**	209,932
15	6	10	**214,252**	215,685	220,529	**214,252**	**214,252**	**214,252**	**214,252**	214,252
16	6	10	**212,588**	214,657	216,291	**212,588**	**212,588**	**212,588**	**212,588**	212,588
17	15	5	485,489	511,854	504,759	**480,453**	**480,453**	480,496	**480,453**	480,453
18	15	5	490,791	507,694	514,718	**484,761**	**484,761**	**484,761**	**484,761**	484,761
19	15	5	494,219	518,461	516,063	489,059	**488,748**	489,430	**488,748**	488,748
20	15	5	490,945	514,242	508,532	484,446	484,446	484,446	484,446	484,405
21	15	5	493,573	512,834	515,599	488,044	487,822	488,054	489,438	487,722
22	15	5	490,735	513,763	509,384	486,898	**486,493**	**486,493**	486,728	486,493
23	15	5	492,301	512,722	512,508	486,855	486,819	487,698	487,370	486,779
24	15	5	496,457	521,116	514,839	491,589	**490,812**	491,237	491,080	490,812
25	15	10	990,646	1,047,596	1,055,536	**978,848**	**978,848**	979,977	980,593	978,848
26	15	10	990,023	1,037,580	1,061,940	978,324	978,523	979,002	978,288	978,065
27	15	10	993,861	1,056,185	1,073,603	983,007	981,191	981,528	981,725	978,027
28	15	10	986,403	1,026,789	1,060,034	972,432	**971,740**	974,389	974,006	971,740
29	15	10	988,881	1,033,591	1,064,692	**976,811**	977,856	978,876	977,260	976,811
30	15	10	977,384	1,028,606	1,066,370	969,443	968,466	971,987	968,362	967,617
31	15	10	991,393	1,043,823	1,066,617	979,166	**978,946**	981,114	979,867	978,946
32	15	10	993,981	1,048,853	1,068,216	**983,486**	984,023	984,865	984,598	983,486
33	30	5	590,200	611,794	632,737	575,288	**575,028**	575,813	575,313	575,028
34	30	5	581,043	611,873	647,585	569,403	**568,057**	568,950	569,643	568,057
35	30	5	584,195	611,664	642,295	573,214	**572,478**	573,414	574,622	572,478
36	30	5	577,629	611,766	634,626	566,655	566,977	566,387	568,524	566,248
37	30	5	571,133	604,564	639,693	557,761	556,269	557,657	**556,257**	556,257
38	30	5	578,039	606,010	637,620	566,715	**565,004**	566,761	565,691	565,004
39	30	5	581,913	607,134	640,482	567,230	568,097	567,859	568,229	567,131
40	30	5	587,653	620,183	635,776	575,345	**573,193**	575,326	574,038	573,193
41	30	10	1,199,376	1,228,411	1,362,513	1,160,830	1,160,949	1,159,188	**1,158,684**	1,158,684
42	30	10	1,200,300	1,231,978	1,379,640	1,162,119	1,159,629	1,160,299	**1,157,547**	1,157,547
43	30	10	1,191,673	1,231,829	1,365,024	1,157,832	**1,153,389**	1,156,093	1,154,147	1,153,389
44	30	10	1,177,912	1,227,413	1,367,130	1,145,004	**1,143,000**	1,144,305	1,143,331	1,143,000
45	30	10	1,163,035	1,215,256	1,356,860	1,129,705	1,125,260	1,125,039	**1,122,196**	1,122,196
46	30	10	1,178,097	1,221,356	1,372,513	1,144,154	1,142,579	1,144,254	1,143,452	1,141,344
47	30	10	1,185,496	1,212,273	1,382,799	1,148,459	1,144,875	1,147,985	1,149,764	1,140,744
48	30	10	1,193,189	1,245,423	1,383,610	1,166,727	1,164,198	**1,160,420**	1,166,223	1,160,420
# of BEST			11	5	2	20	31	19	22	
AVG DEV %			1.38	4.52	8.19	0.13	0.05	0.12	0.10	

#ofBEST : Number of times best is found
AVG DEV%: Average deviation percentage from best known solution
SymEA : Chang (2002); NLGA : Balakrishnan and Cheng (2000);
CONGA : Conway and Venkataramanan (1994)

Comparison of our algorithm with the solution methods from the literature is given in Tables 3.3 and 3.4. After updating the best known solutions using the ones found with different versions of our algorithm, SALAB-R has an average deviation of 0.05% from the best known solutions. According to the tables, SALAB-R outperforms other strategies since it finds the best solution for 31 of the 48 problems. SAII of McKendall et al. (2006) follows SALAB-R with 23 best results and an average deviation of 0.10%. With a Pentium IV 1.8 GHz PC, SALAB-R solves a 30 department 10 period problem in 7.64 minutes on the average. For reference, this time is 7.8 minutes for McKendall et al. (2006) on a Pentium IV 2.4 GHz PC.

When we compare different versions of our algorithm in terms of the new best solutions found, SA*, SALAB-R, SALAB-1 and SALAB-2 solutions are better than the best known solutions for 3, 11, 2 and 4 problems, respectively. All together, they found new best solutions for 18 of the 48 problem instances.

CHAPTER 4

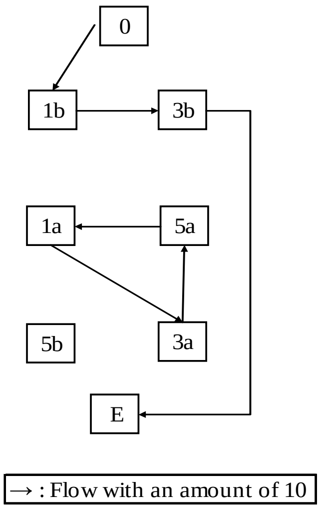
DYNAMIC DISTRIBUTED FACILITY LAYOUT PROBLEM (DDFLP)

As mentioned in Chapter 2, Lahmar and Benjaafar (2005) study the multi-period distributed layout problem or Dynamic Distributed Facility Layout Problem (DDFLP). Given the interdepartmental material flow matrix and relocation costs of department duplicates, this problem makes the decisions of department duplicate-to-location assignments and material flow distribution among department duplicates over time periods. In this chapter, we first modify the DDFLP formulation of Lahmar and Benjaafar (2005) given in Equations 2.11 to 2.18. We improve upon some deficiencies of the model and add the empty travel cost of the material handling system (MHS) to the objective function. We then adapt SALAB-R for solving DDFLP. Finally, we describe the model developed for simulating the DDFLP solutions.

Flow Balance Constraints

Lahmar and Benjaafar's flow balance constraints (given in Equations 2.14 and 2.15) may result in cycles in the flow allocation network, if a part must visit the same department more than once according to its processing route. According to the example given in Figure 4.1, parts visiting department 3 for the first time must go to department 5. On the other hand, parts visiting 3 for the second time must go to exit (E). Lahmar and Benjaafar's flow balance constraints are not sufficient to satisfy these conditions and may result in undesirable flow allocations when the same department is visited more than once. Moreover this may also cause a cycle in the network. For example, all the parts coming to duplicate 3b for the first time are directly sent to exit after processed and a cycle is observed within duplicates 1a, 3a and 5a.

It is possible to use additional constraints for cycle elimination. Instead of that, for simplification, we assume that the same pair of stations cannot exist more than once in the route (for the above example, pair 1-3 exists twice, which violates this assumption). However, we allow multiple visitations of the same department in a route provided that it is succeeded by a different department each time. With this assumption, we propose new balance constraints with which, for every department on the route, all the flow coming from the duplicates of the preceding department exactly go to the duplicates of the succeeding department. This constraint set is similar to the one in the integrated machine allocation and layout problem model of Urban et al. (2000).



Material Flow Between Departments

	0	1	3	5	E
0		10			
1			20		
3				10	10
5		10			
E					

Material Flow Between Department Duplicates

	0	1a	1b	3a	3b	5a	5b	E
0			10					
1a				10				
1b					10			
3a						10		
3b								10
5a		10						
5b								
E								

Figure 4.1. A product with 10 units of demand, and process route 0-1-3-5-1-3-E, where each department has two duplicates (a and b).

According to our assumption, there must be exactly one succeeding department for the flow coming from 1 to 3, and there must be exactly one preceding department for the flow going to 3 from 1. Hence, the pair 1-3 cannot be repeated in the route, but department 1 can be visited twice as in, for example, 0-1-3-5-1-4-E.

Relocation Cost

Relocation decisions involving exchange of duplicates of the same department type do not require a rearrangement in the distributed layout problem. For example, in Figure 4.2, instead of exchanging the locations of department duplicates 1a and 1b of the same type in consecutive periods, only the flow allocated to them can be exchanged without any relocation. Therefore, department duplicate exchanges do not always incur relocation cost. We use a different formulation in which relocation cost is incurred only if different types of department duplicates are assigned to the same location in consecutive periods.

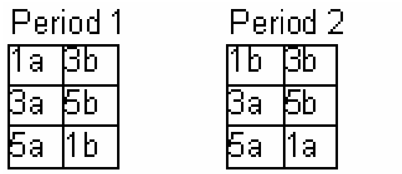


Figure 4.2. A case where relocation of department duplicates is not necessary

4.1 DDFLP Formulation

Assumptions

- The same department cannot be visited consecutively in a processing route, i.e. a 3-3 pair is not allowed.

- A department pair may exist only once in a processing route, i.e. a 1-3 pair is not allowed twice.

- Rearrangement cost is independent of how far the machines are moved.
- There are no common entrance and exit points located in the layout. Incoming parts enter the system at the first department and outgoing parts exit the system at the last department they have to visit. Hence, entrance-to-first station and last station-to-exit trips are ignored, and only interdepartmental trips of the MHS are considered.

- If there are multiple transporters, one of them is selected at random for a transport request.

- Incoming parts (unit loads) have a constant process time with no setup on each department they visit.

Inputs (Parameters)

T: number of time periods

N: number of department types (0 is the entrance)

N_i: number of duplicates for department type i

P: number of product types

M: number of locations

L_p: length of the processing route of product p in terms of the number of departments visited

$R_{p,l}$: department type in the l^{th} position of the processing route of product p (position 0 is the entrance and position L_p+1 is the exit)

V_{ijpt} : total flow volume due to product p between department duplicates of type i and department duplicates of type j in period t

VT_t : total flow (load) of the MHS in period t ($VT_t = \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ijpt}$)

d_{kl} : travel distance between location k and location l

t_{nip} : processing time per unit load of product type p at department duplicate n of type i

c_{kl} : cost (per unit distance) of transporter movement from a department located at k to a department located at l

C_{ni} : capacity (available processing time) of duplicate n of department type i

r_i : cost of relocating a duplicate of department type i

Decision Variables

$$x_{nikt} : \begin{cases} 1 & \text{if } n^{th} \text{ duplicate of department type } i \text{ is assigned to location } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

v_{nimjpt} : volume of flow due to product p between n^{th} duplicate of department i and m^{th} duplicate of department j in period t

$$y_{ikt}: \begin{cases} 1 & \text{if department type } i \text{ is assigned to location } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

Objective Function

Lahmar and Benjaafar's (2005) mixed integer programming formulation accounts only for the full travel cost. The share in cost of each pair of duplicates of department types *i* and *j* is determined by decision variable v_{nimjpt}. To include the cost of empty travel, we can write the objective function as follows.

$$\begin{aligned} \text{Min} \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{n=1}^{N_i}\sum_{j=1}^{N}\sum_{m=1}^{N_j}\sum_{k=1}^{M}\sum_{l=1}^{M} & \frac{\overbrace{\sum_{p=1}^{P}\sum_{q=1}^{N}\sum_{r=1}^{N_q} v_{rqnipt} \sum_{p=1}^{P}\sum_{v=1}^{N}\sum_{z=1}^{N_v} v_{mjzvpt}}^{E}}{VT_t} x_{nikt} x_{mjlt} c_{kl} d_{kl} \\ & + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{n=1}^{N_i}\sum_{j=1}^{N}\sum_{m=1}^{N_j}\sum_{k=1}^{M}\sum_{l=1}^{M} v_{nimjpt} x_{nikt} x_{mjlt} c_{kl} d_{kl} \\ & + \sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{k=1}^{M} y_{ikt}(1 - y_{ik(t+1)}) r_i \end{aligned} \tag{4.1}$$

First term represents the empty travel cost of MHS based on the Equation 2.8 by Johnson (2001). This approximation is valid for a single transporter or when one of the transporters is selected at random.

$$\text{Expected number of trips from } i \text{ to } j = \frac{f_i^{\leftarrow}}{f}\frac{f_j^{\rightarrow}}{f} f = \frac{f_i^{\leftarrow} f_j^{\rightarrow}}{f}$$

Here, expression *E* gives the total number of empty trips in period *t* from n^{th} duplicate of department *i* to m^{th} duplicate of department *j*. The probability that the transporter is at department duplicate *ni* is defined as the sum of flows from all *rq* duplicates to *ni*, divided by the total flow volume. The number of delivery requests issued by *mj* is defined as the sum of flows from *mj* to all *zv* duplicates. The product of these two gives the number of empty trips from *ni* to *mj*. Second term, which is the same as in Lahmar and Benjaafar's (2005) formulation, represents the full travel cost of MHS. Last term represents the rearrangement cost of department duplicates incurring only if different types of departments are placed at the same location in consecutive periods.

Mathematical Model

$$\text{Min}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{n=1}^{N_i}\sum_{j=1}^{N}\sum_{m=1}^{N_j}\sum_{k=1}^{M}\sum_{l=1}^{M}\Bigl(\sum_{p=1}^{P}v_{nimjpt}+\frac{\sum_{p=1}^{P}\sum_{q=1}^{N}\sum_{r=1}^{N_q}v_{rqnipt}\sum_{p=1}^{P}\sum_{v=1}^{N}\sum_{z=1}^{N_v}v_{mjzvpt}}{VT_t}\Bigr)x_{nikt}x_{mjlt}c_{kl}d_{kl} \qquad (4.2)$$
$$+\sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{k=1}^{M}y_{ikt}(1-y_{ik(t+1)})r_i$$

s.t.
$$\sum_{i=1}^{N}\sum_{n=1}^{N_i}x_{nikt}=1 \qquad \forall k,t \qquad (4.3)$$

$$\sum_{k=1}^{M}x_{nikt}=1 \qquad \forall i,n,t \qquad (4.4)$$

$$\sum_{n=1}^{N_i}x_{nikt}=y_{ikt} \qquad \forall i,k,t \qquad (4.5)$$

$$\sum_{m=1}^{N_j}\sum_{n=1}^{N_i}v_{nimjpt}=V_{ijpt} \qquad \forall i,j,p,t \qquad (4.6)$$

$$\sum_{n=1}^{N_{R_{p,l-1}}}v_{nR_{p,l-1}mR_{p,l}pt}=\sum_{r=1}^{N_{R_{p,l+1}}}v_{mR_{p,l}rR_{p,l+1}pt} \qquad \forall l,m,p,t \qquad (4.7)$$

$$\sum_{p=1}^{P}\sum_{i=0}^{N}\sum_{n=1}^{N_i}v_{nimjpt}t_{mjp}\le C_{mj} \qquad \forall j,m,t \qquad (4.8)$$

$$x_{nikt}\in\{0,1\} \qquad \forall i,n,k,t \qquad (4.9)$$

$$y_{ikt}\in\{0,1\} \qquad \forall i,k,t \qquad (4.10)$$

$$v_{nimjpt}\ge 0 \qquad \forall i,n,j,m,p,t \qquad (4.11)$$

Objective function (4.2), which is equivalent to (4.1), includes total (full and empty) MHS traveling cost and relocation cost. Constraints (4.3) and (4.4) ensure one-to-one assignment of department duplicates and locations. Constraints (4.5) are used to find the department type of the duplicate placed at a location. Since entrance (i = 0) and exit ($i = N + 1$) are not located in the layout, constraints (4.3), (4.4) and (4.5) consider all departments except the entrance and exit. Constraints (4.6) equate the flow between duplicates of type i and type j to total flow between types i and j. Constraints (4.7) are similar to the ones used in the integrated machine allocation and layout problem model of Urban et al. (2000). According to these constraints, for every department on a route, all the flow coming from the duplicates of the preceding department exactly goes to the duplicates of the succeeding department. Constraints (4.8) ensure that the workload (in terms of time) of each department duplicate (calculated from the incoming load) does not exceed its capacity. Constraints (4.9) and (4.10) define the location assignment variables as binary and constraints (4.11) ensure that all flows are positive. Flow constraints (4.6), (4.7), (4.8) and (4.11) consider entrance (i = 0) and exit (i = N + 1) points as well as all departments since flow enters the system through the entrance and leaves the system through the exit. Our modifications of Lahmar and Benjaafar's (2005) formulation are in (4.2), (4.5), (4.7) and (4.10).

Limited capacities of department duplicates can cause the objective value to get worse. On the other hand, unbalanced utilization of the same type of department duplicates may have a negative impact on congestion levels and throughput times at the more utilized departments (Benjaafar and Sheikzadeh, 2000). Therefore an equal capacity strategy is implemented for duplicates of the same department type in the experimental runs of Chapter 5.

4.2 Solution Approach for DDFLP

The proposed model is a nonlinear mixed integer problem. If the flow allocations are given, then the problem reduces to DFLP, which is a quadratic mixed integer problem (P1). If the location assignments are given and only the full trips of MHS are considered, the problem can be decomposed into T independent linear problems (P2), which can be solved by a standard LP solver such as CPLEX.

Since the proposed SALAB-R algorithm performs well for DFLP, it is modified for DDFLP. Empty trips, department duplicates and flow allocations are added to the algorithm. Cost function of SALAB is modified to include the empty travel cost and to consider department duplicates for the relocation cost. A similar neighborhood definition is used in which instead of a department, a department duplicate is selected for relocation. Flow allocation can be solved independently for known department duplicate-to-location assignments. Hence, when the algorithm updates the best solution, flow allocation problems (P2s) are solved according to the location decisions in the best solution. P2 is solved independently for every period only with the full travel cost objective and resulting flow allocation decisions are always accepted. We ignore the empty travel cost term only when solving P2s since this term is still nonlinear even after eliminating the location decisions. Hence, our solutions are suboptimal in terms of the total travel cost. However, we still accept the new flow allocation as a way of diversification. The new algorithm, named DSALAB is given in Figures 4.3 and 4.4.

START

Initialization:
    select initial solution S_0, initial temperature t_0, a temperature reduction function T, inner loop iteration limit A.
    set current temperature $t = t_0$, out_loop_count = 0, bestsol = S_0
    determine material flow allocation v for bestsol by solving P2 for each period
    set bestflow = v, bestcost = $f(S_0,v)$

outer loop: repeat while stopping condition is not true
    increase out_loop_count by 1
    iteration_count = 0
    inner loop: repeat while iteration_count is less than A
        increase iteration_count by 1
        randomly select department duplicate n, location l and period p
        execute *NeighborhoodMove*(n,l,p) to generate solution S from neighborhood $N(S_0)$
        If $\Delta = f(S,v) - f(S_0, v) < 0$
            $S_0 = S$,
            if $f(S_0, v)$ < bestcost then
                bestcost = $f(S_0, v)$
                bestsol = S_0
                bestflow = v
                determine new material flow allocation v by solving P2 for each period
                if $f(S_0, v)$ < bestcost then bestcost = $f(S_0, v)$ and bestflow = v
            endif
            execute LAB *
        else
            generate r uniformly distributed in the range [0,1)
            if $r < e^{-\Delta/t}$ then $S_0 = S$ and execute LAB *
        endif
    endwhile
    set current temperature level $t = T(t_0)$
endwhile

END
* LAB is the Look Ahead / Look Back procedure given in Figure 4.4

Figure 4.3. DSALAB Algorithm

```
LAB: for department duplicate n, location l and period p

          START
          period_count = p+1
          repeat while period_count  is less than or equal to the last period
                    increase iteration_count by 1
                    execute NeigborhoodMove(n,l,period_count) to obtain the new layout plan S
                    calculate Δ = f(S,v) – f(S0, v)
                    if Δ < 0
                              S0 = S
                              if f(S0, v) < bestcost  then
                                        bestcost = f(S0, v), bestsol = S0, bestflow = v
                                        determine new material flow allocation v by solving P2 for
                                        each period
                                        if f(S0, v) < bestcost then bestcost = f(S0, v) and bestflow = v
                              endif
                    else
                              generate r uniformly in the range [0,1)
                              if r < e^(-Δ/t) then  S0 = S, else go to NEXT
                    endif
                    increase period_count by 1
          endwhile

NEXT:     period_count = p -1
          repeat while period_count  is greater than or equal to the first period
                    increase iteration_count by 1
                    execute NeigborhoodMove(n,l,period_count) to obtain the new layout plan S
                    calculate Δ = f(S,v) – f(S0, v)
                    if Δ < 0
                              S0 = S
                              if f(S0, v) < bestcost  then
                                        bestcost = f(S0, v), bestsol = S0, bestflow = v
                                        determine new material flow allocation v by solving P2 for
                                        each period
                                        if f(S0, v) < bestcost then bestcost = f(S0, v) and bestflow = v
                              endif
                    else
                              generate r uniformly in the range [0,1)
                              if r < e^(-Δ/t) then  S0 = S, else go to END
                    endif
                    decrease period_count by 1
          endwhile
          END
```

Figure 4.4. Look Ahead/Back procedure LAB for DSALAB

4.3 Simulating the Dynamic Distributed Facility

Mathematical model calculates the full MHS cost, empty MHS cost and relocation cost. A simulation model can be used to predict these measures for a stochastic system, and these can be compared with the approximate measures obtained from the mathematical model. Moreover, other operational metrics can also be estimated by simulation.

We developed a simulation model of the dynamic distributed facility in SIMAN. The system is simulated according to the location and flow allocation results of the mathematical model. Full and empty utilizations of transporter, department or machine utilizations, WIP accumulated by the MHS and total WIP are estimated using the model. We now briefly describe the model characteristics.

Inputs

- Number of time periods, T
- Number of department types, N
- Number of duplicates for department type i, N_i
- Number of product types, P
- Processing route of each product type
- Distance matrix for distances between department duplicates located according to the DSALAB solution
- Total demand per period
- Demand mix for each period
- Material flow between department duplicates for each period as a function of the demand mix and flow allocation according to the DSALAB solution
- Processing time for each department type
- Period length, PL

Assumptions

- Demand is generated by an exponentially distributed interarrival time.
- Processing times of machines are exponentially distributed.
- There is only one transporter in the system.
- Transporter moves with a constant speed.

Simulation Model

The same model file is used in all experiments by only changing the experiment file of the SIMAN model.

In this model, different types of products are processed according to their processing routes. There exist $\sum_{i=1}^{N} N_i$ stations in the model for representing the department duplicates located in each period according to the DSALAB solution. Specifically, the model does not explicitly use the locations of departments; instead it uses the distance matrix of departments generated from their locations. Demand mix and distance matrix change over time periods.

Demand arrives according to an exponential distribution. Mean interarrival time is calculated by dividing the period length with total period demand. An arrival is assigned a certain product type by a discrete distribution according to the demand mix of the current period. Then the product follows its processing route. A product can visit any duplicate of a department in its processing route according to a discrete distribution. In this distribution, the probability of selecting a certain duplicate is determined for each period by the flow allocation decisions in the DSALAB solution. Probability that a product of type *p* goes from n^{th} duplicate of department *i* to m^{th} duplicate of department *j* that comes next in its route is found as:

$$\frac{v_{nimjpt}}{\sum_{m=1}^{N_j} v_{nimjpt}}$$

A transporter is used for carrying products between departments. An arriving product directly goes to its first processing department without using the transporter (transporter is not used for incoming and outgoing products; it is only used for transportation between department duplicates). When a product arrives at a department duplicate, it first waits in the input buffer of the station until it is processed. After processed, it waits in the output buffer, which is a common queue for the transporter. Every product processed in any department duplicate waits in this common transporter queue to be transported. Transporter serves the products according to the first come first served queue discipline. Because of the limitations of SIMAN, it is not possible to change the distance matrix of a transporter during the simulation run. Therefore, different transporters are used for each period and, at the end of the period, the products waiting in the transporter queue are transferred to the transporter queue of the next period.

Time of interperiod changes is determined according to the number of arrivals and the number of parts produced. When the number of arrivals reaches the total demand of a period, the model starts to assign product type of subsequent arrivals according to the demand mix of the next period. When the number of departures reaches the demand of a period, then the locations of departments are changed. This change means that the model starts to use the distance matrix of the next period. There is a transition time from the current period to the next period. Products of the next period begin to arrive before all products of the current period are finished and depart, meaning that the production facility continues to produce products of both periods. This is unavoidable; however it is important to have a small transition time to approximate the results of the mathematical model with small deviation. Therefore, the period length should be carefully decided. It should be long enough to avoid overutilization of department duplicates and the transporter, and result in small transition times. It should also be short enough to avoid underutilization of the system and to allow observation of WIP accumulation.

As far as the output analysis is concerned, this is neither a terminating nor a steady-state simulation. Therefore, we simulate the first and the last periods twice and truncate the extra periods from the beginning and the end of the run. This is because we do not want to start with an empty facility and end up with an empty facility. We calculate the performance measures using only the original periods. As an example, an eight-period problem of the mathematical model becomes a ten-period problem in the simulation run and the statistics are collected for periods 2-9.

CHAPTER 5

EXPERIMENTAL RESULTS FOR DDFLP

After integrating DSALAB and simulation, randomly generated problems are solved for several demand amounts per period, demand mix levels over periods, department duplication levels and machine relocation costs. Moreover, varying period length values are used to prevent the system from having long interperiod transition times and overutilization in the simulation runs.

5.1 Experimental Conditions

We have generated 30 problems and applied the DSALAB optimization and simulation to each problem for 64 experimental settings. Procedures of experimentation are detailed in APPENDIX C including some specifics of problem generation and implementation of optimization and simulation.

Problem Generation

We have generated 30 problems at random. Each problem has eight periods, eight product types and six department types. A complete list of problem parameters is given in Table 5.1.

Product types are characterized by their processing routes, hence product type and route are used as synonyms. Length of the processing route of each product (the number of department visitations) is generated uniformly between 2 and 8. Departments in a processing route are selected randomly with the assumptions that the same department cannot be visited consecutively and a department pair may exist only once in a processing route. The first four products and the second four products differ by the usage frequency of departments in their processing routes. The most frequently occurring departments in the first four routes become the least frequently occurring ones in the second four. Detailed problem generation procedure is given in Figure 5.1. An example of route generation is given in Table 5.2.

Table 5.1. Details of Parameters

Parameter	Value
Number of periods, T	8
Number of department types, N	6
Number of machines for department type i, N_i	8 for $i = 1,\ldots, N$
Number of product types, P	8
Number of locations, M	48 (8x6 grid) (see Table 5.5 and Figure 5.2 for different department duplication levels)
Processing route length for part p, L_p	Generated uniformly between 2 and 8 for $p = 1,\ldots, P$
Processing route for part p, $R_{p,l}$	Generated by randomly selecting one of the N department types for $l = 1,\ldots, L_p$, $p = 1,\ldots, P$
Process time at department i, t_i (t_{nip} is reduced to t_i assuming all parts have the same process time in all duplicates of department type i)	Generated uniformly between 10 and 100, generated value is used as the deterministic process time in the mathematical programming model and as the expected value of the exponentially distributed process time in simulation
Cost per unit distance of transporter move, c_{kl}	1 for all location pairs k, $l = 1,\ldots, M$
Distance between grid locations k and l, d_{kl}	Rectilinear between centers of grids where grid size is taken as 1x1 for all pairs k, $l = 1,\ldots, M$ (see Table 5.5 and Figure 5.2 for different department duplication levels)
Machine relocation cost, r (r_i is reduced to r assuming all machines have the same cost)	Experimental factor
Total demand per period, D	Experimental factor
Demand mix level, ML	Experimental factor
Duplication level, DL	Experimental factor
Total flow volume of product p between department types i and j, V_{ijpt}	Calculated as a function of D, ML and $R_{p,l}$ (see APPENDIX C)
Capacity of a duplicate of department type i, C_i (C_{ni} is reduced to C_i assuming all duplicates of a department are identical)	Calculated as a function of v_{ijpt} and t_i (see APPENDIX C)

START
for p = 1 to 4
    generate route length, L_p, uniformly between 2 and 8 for product p.
    generate departments of processing route p, $R_{p,l}$, by preventing occurrence of the same department pair in the route and occurrence of the same department consecutively.
endfor
calculate the usage frequency of departments in the routes 1-4.
for p = 5 to 8
    copy route p-4
    change the departments in the processing route according to the usage frequency. (Most frequently occuring department is replaced by the least used, second most used by the second least used, and so on)
endfor
for i = 1 to N
    generate process time of department i, t_i, uniformly between 10 and 100.
endfor
END

Figure 5.1. Detailed problem generation procedure

Table 5.2. An example for route generation

Part Type, p	L_p	$R_{p,l}$
1	7	{5, 6, 1, 2, 5, 3 ,2}
2	5	{1, 6, 3, 1, 3}
3	7	{6, 3 ,2 ,1 ,3 ,1 ,2}
4	2	{3, 1}
5	7	{1, 2, 5, 6, 1, 4, 6}
6	5	{5, 2, 4, 5, 4}
7	7	{2, 4, 6, 5, 4, 5, 6}
8	2	{4, 5}

Usage frequencies of departments according to the first four routes in Table 5.2 are given in Table 5.3. To generate route 5 from route 1, we replace department 3 in route 1 with department 4, 1 with 5, and 2 with 6. Routes 6, 7 and 8 are generated similarly from routes 2, 3 and 4, respectively. The motivation is to create a dynamic environment where, as the product mix changes over periods, workload shifts from certain departments to others.

Table 5.3. Usage frequencies of departments according to the first four routes

Department type	3	1	2	6	5	4
Usage frequency	6	6	4	3	2	0

Processing times for machines of different department types are generated uniformly between 10 and 100 for each of six department types.

Computation of flow volume between departments (V_{ijpt}) and department capacities (C_i) are given in APPENDIX C. Capacity of a department is not changed over periods and assigned as the workload of the period in which the department is maximally used (hence a slack capacity exists in other periods). This capacity is divided equally among the department duplicates. Capacities are scaled for different duplication levels since the number of machines in department duplicates varies according to this experimental factor.

Experimental Factors

Factors and their levels are:

- Relocation cost per machine (r): 1, 15, 50, ∞
- Total demand per period (D): 200, 400 unit loads
- Demand mix level (*ML*): ML1, ML2
- Duplication level (*DL*): DL1, DL2, DL4, DL8

Hence, 64 combinations are solved for 30 problems.

We use different machine relocation costs to facilitate large, medium and small number of machine relocations over periods. Moreover, a static distributed facility layout problem (SDFLP) is created with $r = \infty$. SDFLP assumes that the location and flow allocation decisions do not change over periods, and distribution of department duplicates is the only mechanism to respond to dynamic changes.

By using two different values for total period demand but the same number of periods and period length for the simulation runs, we can analyze both the more utilized and less utilized systems.

With the two demand mix levels and the processing route generation procedure, we can generate a dynamic environment in which demand mix and utilization of departments change over periods. Demand mix percentages over periods are given in Table 5.4. For ML1, only the first four product types are produced in the first period with shares 40, 30, 20, 10 percent of the total demand. The shares of these products in the total demand decrease over the periods, while the shares of the last four products increase starting with the second period. Only the last four product types are produced in the last period. Note that shares of all eight product types add up to 100 percent for each period. With the help of the processing route generation, departments having more workload in the early periods will have less workload in the late periods and vice versa. In ML2, again only the first four product types are produced in the first period. Their shares decrease and shares of the last four products increase starting with the second period. However, all eight product types have equal shares in the last period. By this way, departments having more workload in the early periods will have less workload in the late periods, but workload of all departments will be nearly the same in the last period.

Table 5.4. Demand mix percentages for ML1 and ML2 for 8 products over 8 time periods

ML1	t = 1	t = 2	t = 3	t = 4	t = 5	t = 6	t = 7	t = 8
p = 1	40.00	34.29	28.57	22.86	17.14	11.43	5.71	0.00
p = 2	30.00	25.71	21.43	17.14	12.86	8.57	4.29	0.00
p = 3	20.00	17.14	14.29	11.43	8.57	5.71	2.86	0.00
p = 4	10.00	8.57	7.14	5.71	4.29	2.86	1.43	0.00
p = 5	0.00	5.71	11.43	17.14	22.86	28.57	34.29	40.00
p = 6	0.00	4.29	8.57	12.86	17.14	21.43	25.71	30.00
p = 7	0.00	2.86	5.71	8.57	11.43	14.29	17.14	20.00
p = 8	0.00	1.43	2.86	4.29	5.71	7.14	8.57	10.00

ML2	t = 1	t = 2	t = 3	t = 4	t = 5	t = 6	t = 7	t = 8
p = 1	40.00	36.07	32.14	28.21	24.29	20.36	16.43	12.50
p = 2	30.00	27.50	25.00	22.50	20.00	17.50	15.00	12.50
p = 3	20.00	18.93	17.86	16.79	15.71	14.64	13.57	12.50
p = 4	10.00	10.36	10.71	11.07	11.43	11.79	12.14	12.50
p = 5	0.00	1.79	3.57	5.36	7.14	8.93	10.71	12.50
p = 6	0.00	1.79	3.57	5.36	7.14	8.93	10.71	12.50
p = 7	0.00	1.79	3.57	5.36	7.14	8.93	10.71	12.50
p = 8	0.00	1.79	3.57	5.36	7.14	8.93	10.71	12.50

Four levels of department duplication are DL1, DL2, DL4 and DL8. There are 8 machines in each department type (a total of 48 machines with 6 department types) for all cases. In DL1, there is only one department of each type, and this department contains all eight parallel identical machines. In DL2, departments in DL1 are disaggregated into two duplicates each having four machines. In DL4, there are four duplicates of each department type, each having two machines. DL8 represents a completely distributed layout where each machine is regarded as a department duplicate. Department duplicate relocation cost and capacity per department duplicate are doubled from DL8 to DL4, DL4 to DL2, and DL2 to DL1 since the number of machines per duplicate is doubled as the number of duplicates decreases.

We use a grid layout with 8×6 = 48 equal sized grids for locating machines in all duplication levels. Details for varying duplication levels are given in Table 5.5 and Figure 5.2. Size of a duplicate in X and Y coordinates is given in Duplicate Size column of Table 5.5 in terms of the number of grids. For varying duplication levels, Duplicate Relocation Cost column shows the scaling of machine relocation cost considering the number of machines per duplicate. DDFLP column contains the X and Y scales of the optimization problem, where the size of the DDFLP problem is X×Y departments and 8 periods.

Distances between department duplicates are scaled for varying duplication levels to have an equal sized 8×6 facility for all duplication levels represented in Figure 5.2. We use center-to-center rectilinear distances between department centers, which are given in the last column of Table 5.5.

Table 5.5. Characteristics of department duplicates

	Duplicate Size		Duplicate Relocation Cost			DDFLP		Distance Between Centers of Adjacent Departments	
	X	Y	r = 1	r = 15	r = 50	X	Y	X	Y
DL1	4	2	8	120	400	2	3	4	2
DL2	2	2	4	60	200	4	3	2	2
DL4	2	1	2	30	100	4	6	2	1
DL8	1	1	1	15	50	8	6	1	1
r : relocation cost per machine									

DL1

Y

A	A	A	A	D	D	D	D
A	A	A	A	D	D	D	D
C	C	C	C	B	B	B	B
C	C	C	C	B	B	B	B
E	E	E	E	F	F	F	F
E	E	E	E	F	F	F	F

X

DL4

Y

A	A	C	C	E	E	C	C
D	D	B	B	D	D	A	A
F	F	E	E	A	A	B	B
E	E	D	D	F	F	D	D
B	B	C	C	A	A	F	F
C	C	F	F	E	E	B	B

X

DL2

Y

A	A	C	C	A	A	D	D
A	A	C	C	A	A	D	D
D	D	B	B	E	E	C	C
D	D	B	B	E	E	C	C
F	F	E	E	F	F	B	B
F	F	E	E	F	F	B	B

X

DL8

Y

A	C	D	A	E	C	B	E
E	B	F	D	F	B	D	C
A	D	B	F	E	D	E	F
E	C	F	E	A	B	C	D
B	E	D	F	B	C	A	F
A	B	A	C	F	A	D	C

X

Figure 5.2. Grid layout for duplication levels where letters A-F represent machines of different department types and rectangles represent department duplicates

DSALAB Settings

If we consider the DDFLP size, we have 6 department 8 period problems for DL1. The number of departments is 12, 24 and 48 for DL2, DL4 and DL8, respectively.

Each of the 64 factorial combinations for each of the 30 problems is solved by DSALAB starting from a random initial layout with parameters given in Table 5.6. These parameters are based on our SALAB settings by considering the size of the problems. DL1 uses the 6D10P SALAB settings, DL2 uses the 15D10P SALAB settings and both DL4 and DL8 use 30D10P SALAB settings.

Table 5.6. DSALAB parameters

	α	A	TERMCOND
DL1	0.997	1,440	5,000,000
DL2	0.998	7,500	30,000,000
DL4	0.998	12,000	60,000,000
DL8	0.998	12,000	60,000,000

For the case of $r = \infty$, a static distributed layout (SDFLP) is considered over periods. Flow matrix of a product is found by summing up its flow matrices over all periods. The inner loop iteration limit (A) and terminating condition (TERMCOND) of SA, given in Table 5.6, are divided by the number of periods for SDFLP.

Simulation Settings

Each solution of DSALAB is simulated for 30 replications by the SIMAN model described in Chapter 4. One additional parameter needed for simulation is the period length, which is decided considering the workload in time units. As the processing route length is distributed uniformly between 2 and 8, the expected number of operations is 5 for a product. The expected process time per operation is 55 minutes since the process time is also uniformly distributed between 10 and 100 minutes. With a total demand of 400 unit loads of products per period, the expected workload of a period is 400 x 5 x 55 minutes. This load is randomly distributed among 48 machines hence the period length can be 400 x 5 x 55 / 48 = 2292 minutes. However, there is variability due to random generation of processing routes, interarrival times, product types and process times. Therefore, to prevent the system from overutilization and to avoid long interperiod transition times, twice this period length ($PL = 4583$ minutes) is used in all runs. As this period length still causes bottlenecks and close to 100% utilization of some machines, three times the original period length ($PL = 6875$ minutes) is also tried when the total demand is 400. Transporter velocity is taken as 4 unit distance per unit time, which does not result in overutilization of the transporter for the selected period length values. This velocity is decided by experimentation, with the aim of keeping the transporter utilization relatively low and not allowing it to be a bottleneck resource. The purpose is to isolate the MHS effects so that the effects of demand characteristics and distribution of machines can be observed clearly. In simulating DSALAB solutions, we want the realized period length to be close to the intended value (4583 or 6875), with a small interperiod transition percentage and without underutilization and overutilization of the machines and the transporter.

Implementation details of DSALAB and simulation are given in APPENDIX C.

5.2 Results of DSALAB

Average results for 30 problems are given in Table 5.7 and in APPENDIX D. We report on the three (relocation, empty travel and full travel) cost components and percent improvement of these cost components from the initial solution for all factor combinations. NMR stands for the number of machine relocations. CPU time is given in minutes. According to these results, if the high utilization case (period demand of 400 with period length of 4583) is ignored, DSALAB has an average absolute deviation of 3.32% from the simulation results for the flow cost. This value is 7.48% for the high utilization case. In the high utilization case, since the average interperiod transition times are 35-145 % of the period length, a large portion of the current period's demand is actually produced in the subsequent period(s). That is, the periods overlap to a large extent, and they are not separated as intended. This results in a relatively large deviation of DSALAB results from the simulation results. SDFLP ($r = \infty$) also has a long interperiod transition time but the DSALAB results do not deviate much from the results of simulation in terms of flow cost since the values of decision variables do not change by period. Interperiod transition times are further discussed under the results of simulation runs. With an Intel Dual Core 2.33 GHz PC, it takes approximately 20 minutes to solve the DL8 case, which is a 48 department duplicate and 8 period dynamic distributed facility layout problem. The CPU time is less than two minutes for r = ∞, as this is a single period or static distributed facility layout problem.

Table 5.7. Improvement details and results for DSALAB (average over 30 problems)

DEMAND	ML	DL	IMPROVEMENT % FROM INITIAL RELOCATION	EMPTYFLOW	FULLFLOW	TOTAL	COST RELOCATION	EMPTYFLOW	FULLFLOW	TOTAL	NMR AVG	MAX	CPU AVG
r = 1													
200	1	1	79.51	3.46	20.76	13.28	55.73	24,916.75	24,677.71	49,650.20	55.73	96	0.3
200	1	2	19.90	5.99	26.12	14.83	214.67	26,788.92	16,334.11	43,337.69	214.67	280	3.0
200	1	4	15.21	8.48	31.48	15.87	244.20	27,047.20	9,648.49	36,939.88	244.20	278	10.3
200	1	8	16.05	9.87	27.09	14.33	245.97	27,368.35	7,646.81	35,261.13	245.97	275	19.4
200	2	1	90.49	3.01	19.36	12.34	25.87	24,949.94	24,913.24	49,889.04	25.87	88	0.3
200	2	2	19.90	4.82	25.23	13.78	214.67	27,233.12	16,633.53	44,081.32	214.67	288	3.0
200	2	4	15.81	5.85	30.37	13.83	242.47	27,985.05	10,027.32	38,254.84	242.47	274	10.4
200	2	8	15.80	6.79	27.33	12.18	246.70	28,453.03	7,799.62	36,499.35	246.70	267	19.6
400	1	1	78.24	3.46	20.77	13.11	59.20	49,833.00	49,352.38	99,244.58	59.20	112	0.3
400	1	2	11.54	6.25	25.72	14.75	237.07	53,385.42	32,826.60	86,449.08	237.07	272	3.0
400	1	4	13.91	8.43	31.25	15.76	247.93	54,110.65	19,343.58	73,702.16	247.93	286	10.3
400	1	8	14.96	9.62	27.40	14.22	249.17	54,882.36	15,199.77	70,331.30	249.17	268	19.4
400	2	1	89.41	3.01	19.37	12.14	28.80	49,898.35	49,824.10	99,751.25	28.80	88	0.3
400	2	2	20.70	4.92	25.18	13.81	212.53	54,460.68	33,235.41	87,908.62	212.53	272	3.0
400	2	4	14.05	5.97	30.20	13.84	247.53	55,905.89	20,064.14	76,217.56	247.53	280	10.4
400	2	8	12.81	6.90	27.05	12.16	255.47	56,796.36	15,676.87	72,728.70	255.47	282	19.6
r = 15													
200	1	1	91.47	3.29	20.37	18.30	348.00	24,957.30	24,790.67	50,095.96	23.20	56	0.3
200	1	2	81.09	5.40	24.11	19.00	760.00	26,900.01	16,747.05	44,407.06	50.67	108	3.0
200	1	4	67.45	7.94	29.92	20.25	1,406.00	27,176.61	9,855.20	38,437.82	93.73	168	10.4
200	1	8	66.18	9.09	25.49	18.99	1,486.50	27,583.82	7,799.87	36,870.18	99.10	140	19.5
200	2	1	98.43	2.85	19.01	17.88	64.00	24,989.32	25,008.67	50,061.99	4.27	24	0.3
200	2	2	88.81	4.38	22.98	18.56	450.00	27,356.26	17,084.86	44,891.12	30.00	112	3.0
200	2	4	65.23	5.68	28.06	18.12	1,502.00	28,017.76	10,359.79	39,879.55	100.13	196	10.5
200	2	8	66.71	6.59	24.50	17.16	1,463.00	28,532.42	8,103.58	38,099.00	97.53	199	19.7
400	1	1	89.61	3.37	20.45	15.60	424.00	49,874.79	49,530.29	99,829.07	28.27	64	0.3
400	1	2	62.44	5.72	25.99	16.68	1,510.00	53,646.34	32,740.49	87,896.82	100.67	200	3.0
400	1	4	45.60	8.10	30.78	17.03	2,350.00	54,301.23	19,491.55	76,142.78	156.67	226	10.4
400	1	8	48.04	9.51	26.86	15.99	2,283.50	54,939.89	15,296.19	72,519.58	152.23	203	19.5
400	2	1	95.98	2.95	19.19	14.95	164.00	49,925.23	49,918.19	100,007.42	10.93	32	0.3
400	2	2	62.54	4.31	24.50	15.27	1,506.00	54,722.52	33,533.61	89,762.13	100.40	192	3.0
400	2	4	46.69	5.84	28.83	15.11	2,303.00	55,921.40	20,474.91	78,699.31	153.53	214	10.5
400	2	8	47.06	6.88	26.35	13.98	2,326.50	56,806.03	15,783.60	74,916.13	155.10	194	19.7

Table 5.7. (Continued)

r = 50													
			IMPROVEMENT % FROM INITIAL				COST				NMR		CPU
DEMAND	ML	DL	RELOCATION	EMPTYFLOW	FULLFLOW	TOTAL	RELOCATION	EMPTYFLOW	FULLFLOW	TOTAL	AVG	MAX	AVG
200	1	1	99.12	2.82	18.64	29.30	120.00	25,076.12	25,263.81	50,459.93	2.40	32	0.31
200	1	2	95.77	4.22	22.82	30.76	566.67	27,195.99	17,086.49	44,849.15	11.33	56	2.89
200	1	4	90.79	6.53	25.42	33.05	1,326.67	27,556.77	10,438.66	39,322.10	26.53	56	9.86
200	1	8	88.03	7.49	21.52	32.42	1,753.33	28,034.63	8,185.59	37,973.55	35.07	60	18.38
200	2	1	100.00	2.72	18.85	29.43	0.00	25,025.57	25,061.33	50,086.91	0.00	0	0.30
200	2	2	98.21	3.71	22.61	30.83	240.00	27,552.02	17,174.41	44,966.43	4.80	40	2.87
200	2	4	95.19	4.73	25.54	33.08	693.33	28,241.00	10,668.55	39,602.88	13.87	38	9.97
200	2	8	91.00	5.54	20.85	31.90	1,318.33	28,802.10	8,449.81	38,570.25	26.37	56	18.67
400	1	1	95.88	2.95	19.84	21.69	560.00	50,085.03	49,917.33	100,562.36	11.20	32	0.30
400	1	2	89.90	4.88	25.13	23.23	1,353.33	54,215.00	33,031.97	88,600.31	27.07	60	2.86
400	1	4	77.66	6.99	28.92	23.82	3,216.67	54,939.66	19,998.74	78,155.07	64.33	108	9.91
400	1	8	78.46	8.78	24.07	23.52	3,155.00	55,299.50	15,895.16	74,349.66	63.10	107	18.48
400	2	1	99.61	2.73	18.92	21.55	53.33	50,046.33	50,071.62	100,171.28	1.07	16	0.30
400	2	2	93.33	4.32	22.79	22.44	893.33	54,734.18	34,395.38	90,022.89	17.87	76	2.88
400	2	4	81.60	5.54	27.75	23.15	2,650.00	56,125.76	20,794.82	79,570.58	53.00	102	10.34
400	2	8	79.44	6.21	23.54	21.88	3,011.67	57,184.62	16,411.40	76,607.69	60.23	136	19.34
r = ∞													
200	1	1	-	1.23	16.79	9.80	-	25,650.13	25,392.00	51,042.13	-	-	0.02
200	1	2	-	4.81	22.72	12.70	-	27,475.94	17,162.66	44,638.60	-	-	0.29
200	1	4	-	4.03	27.79	11.74	-	27,553.84	9,864.80	37,418.64	-	-	0.83
200	1	8	-	8.29	20.33	11.36	-	27,800.23	7,947.69	35,747.91	-	-	1.57
200	2	1	-	2.51	16.93	10.47	-	25,141.94	25,061.33	50,203.28	-	-	0.02
200	2	2	-	4.50	26.14	14.14	-	27,493.76	16,730.67	44,224.43	-	-	0.29
200	2	4	-	4.55	30.75	13.25	-	28,040.33	9,972.65	38,012.99	-	-	0.83
200	2	8	-	6.51	24.72	11.34	-	28,450.68	8,061.79	36,512.46	-	-	1.58
400	1	1	-	1.23	16.79	9.80	-	51,300.25	50,784.00	102,084.25	-	-	0.02
400	1	2	-	4.81	22.81	12.74	-	54,951.77	34,266.67	89,218.44	-	-	0.29
400	1	4	-	3.97	27.67	11.65	-	55,161.49	19,772.30	74,933.79	-	-	0.83
400	1	8	-	8.17	20.07	11.20	-	55,657.81	15,970.47	71,628.28	-	-	1.57
400	2	1	-	2.51	16.93	10.47	-	50,283.88	50,122.67	100,406.55	-	-	0.02
400	2	2	-	4.50	25.98	14.08	-	54,985.67	33,528.00	88,513.67	-	-	0.29
400	2	4	-	4.70	30.36	13.22	-	55,979.31	20,073.53	76,052.85	-	-	0.83
400	2	8	-	6.67	24.53	11.41	-	56,817.26	16,176.66	72,993.92	-	-	1.58

Total Cost

Improvement of DSALAB for relocation, full and empty MHS and total cost figures is given in Table 5.7 as the average over 30 problems. Total cost of the random initial layout with optimal flow allocations accounting for only the full travel cost, is improved by 17% on the average. Most improved case is where $r = 50$. DSALAB improves all cost components of the randomly generated initial solutions but improvement percentage of the full travel cost is always higher than that of the empty travel cost. In general, empty travel cost is improved better for ML1 and DL8. Full travel cost is best improved for DL4. Improvement in the relocation cost is larger for DL1 since moving all 8 machines in a single department duplicate has higher impact.

In terms of the cost components, results for different period demands and mix levels are given in Figures D.1 through D.4 in APPENDX D. According to these figures, relocation cost has an insignificant share in the total cost. The highest portion of the total cost is due to empty travel. This cost component slightly increases by duplication level but the rate of increase decreases from DL1 to DL8. DSALAB makes the flow allocation decisions considering only the full travel cost and this may have a negative effect on the empty travel cost while duplication level is increased. Full travel cost decreases by duplication level but this decrease slows down after DL4. While disaggregating large departments into smaller ones, decrease in full travel cost is so large that increase in relocation and empty travel costs become insignificant, resulting in improved total cost. Total cost decreases by 27% from DL1 to DL8 but the least improvement is 3%, from DL4 to DL8 with respect to DL1. Similar patterns are observed for all period demand (D) and demand mix (ML) levels.

A decrease in total cost by department disaggregation is expected since departments are restricted to stay together in DL1, but they can move freely in distributed cases. However, it is experimentally seen that the decrease in total cost slows down after DL4. Hence, we can conclude that DL4 is a sufficient distribution level for our problem instances.

Machine Relocations

Average number of machine relocations is plotted against duplication levels in Figure D.5 for different period demand, demand mix and relocation cost levels. Since the graphics have similar trends, overall averages are given in Figure 5.3 for the three relocation cost levels.

According to Figure 5.3, the number of machine relocations has an increasing trend by duplication level. Since the number of machines per department duplicate decreases by DL, it is easier to relocate a duplicate having only one machine in DL8, compared to a duplicate having eight machines in DL1.

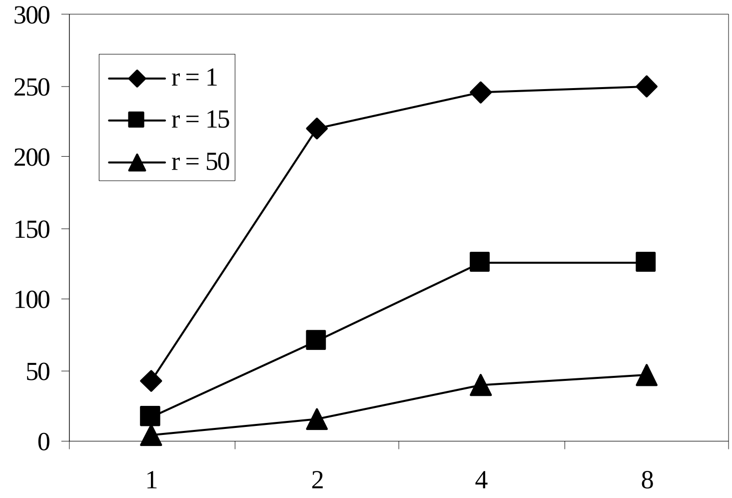


Figure 5.3. Number of machine relocations vs. duplication level

From DL1 to DL8, the number of machine relocations increases by 207, 110 and 43 for relocation costs of 1, 15 and 50, respectively. However results of DL4 and DL8 do not differ much. From DL4 to DL8, the number of machine relocations change by 4, 0 and 7 for relocation costs of 1, 15 and 50, respectively. Low machine relocation cost case has the largest number of relocations as expected. The highest machine relocation cost case, r = 50, results in fewer than 50 machine relocations on the average. It should be noted that relatively low machine relocation costs are used in the experiments to be able to observe machine relocations.

5.3 Results of Simulation Runs

Averages over 30 problems for all experimental factor combinations are given in Tables E.1 through E.4 in APPENDIX E. Respective graphics are given in Figures E.1 through E.5. Since the same pattern is observed for most factor combinations, graphics for overall averages are given in this section. Three utilization levels are shown in the results: 400x2 (period demand is 400 and period length is 4583), 400x3 (period demand is 400 and period length is 6875) and 200x2 (period demand is 200 and period length is 4583).

Selected period length value given above is used to determine the arrival rate in the simulation model, e.g. mean time between arrivals for 400x2 case is found as 4583 / 400. With the constant transporter velocity of 4 unit distance per minute, generated processing times, processing routes and given period length values, actual period length is realized as different from the selected period length. Realized period length is the time interval between the last departure of the previous period's demand and the last departure of the current period's demand. Realized period lengths from the simulation runs are 4600, 4947 and 6903 on the average for 200x2, 400x2 and 400x3, respectively. The motivation for the 400x3 case is that the simulation could not achieve the selected period length value in the 400x2 case, which results in relatively large interperiod transition times. In the 400x2 case, interperiod transition time is between 35% and 145% of the simulation period length whereas it is less than 15% for finite *r* cases of 400x3 and 200x2. 100% interperiod transition percentage means that interperiod transition lasts for a period. This can be interpreted as some of the demand of the current period is produced in the subsequent period. For the total travel cost, DSALAB results do not deviate much from the simulation results having a small interperiod transition time as explained under the results of DSALAB.

We want the realized period length to be close to the selected value (4583 or 6875), with a small interperiod transition percentage and without underutilization and overutilization of machines and the transporter. This is achieved for 400x3 and 200x2 cases with a transporter velocity of 4 unit distance per minute. On the average, interperiod transition is 10% and 12%, maximally used department duplicate utilization is 45% and 54% and transporter utilization is 28% and 38% for 200x2 and 400x3, respectively. However, the highly utilized 400x2 case has interperiod transition percentage of 60%, maximally used department duplicate utilization of 71% and transporter utilization of 57% on the average. Maximally used department duplicate utilization refers to the MAX DEPT UTIL % column in APPENDIX E. It is the average of utilizations of the maximally used duplicates of all department types. Although the average department utilizations are relatively low, they exceed 90% in certain cases.

It is observed that simulation period length values and interperiod transition percentages are larger for infinite *r* (SDFLP) compared to finite *r*. On the average, for finite *r*, it is 9%, 52% and 9% whereas for infinite *r*, it is 13%, 83% and 22% for 200x2, 400x2 and 400x3, respectively. Moreover, for infinite *r*, maximally used department duplicate utilization is 51%, 73% and 60% for 200x2, 400x2 and 400x3, respectively. On the other hand, for finite r, maximally used department duplicate utilization is 43%, 70% and 52% for 200x2, 400x2 and 400x3, respectively. Therefore, congestion in infinite *r* case is higher compared to finite *r* cases.

In terms of the computation time, with a Pentium IV 1.8GHz PC, it takes nearly 7 minutes to simulate the DL8 cases. DL4 simulation lasts approximately 3 minutes and run time of DL2 and DL1 is less than 2 minutes.

Total Transporter Utilization

According to simulation runs, the effect of duplication levels on the total transporter utilization is shown in Figure 5.4. Data points are the averages of 30 problems over 4 relocation cost and 2 demand mix levels. Full and empty travel costs show the same characteristics as full and empty transporter utilizations since cost is a linear function of utilization.

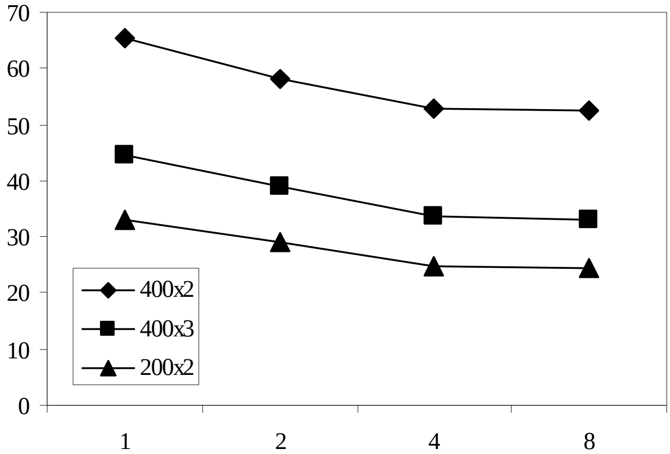


Figure 5.4. Total transporter utilization vs. duplication level

Increasing the duplication level reduces the total transporter utilization but this effect is insignificant after DL4. It reduces by 20%, 26% and 26% from DL1 to DL8 but improvement from DL4 to DL8 with respect to DL1 is 0.3%, 1.8% and 1.9% for 400x2, 400x3 and 200x2, respectively. The same pattern is also observed for individual graphics in Figure E.1, which means that behavior of transporter is mostly dependent on the duplication level. Decreasing trend of total transporter utilization can be explained by the fact that department duplicates are not forced to be grouped in DL8. Hence the less constrained case, DL8, has the least total transporter utilization, resulting in the minimum total travel cost.

However it is experimentally found that there is an insignificant improvement after DL4.

Total Cost

Total cost includes the relocation cost and the travel cost of the transporter according to the simulation runs. In all cases of Figure E.2, increasing the duplication level generally reduces the total cost. Travel cost is a linear function of transporter utilization, and it was found that total transporter utilization does not change much from DL4 to DL8. Since relocation cost increases from DL4 to DL8, disaggregation from DL4 to DL8 becomes insignificant also for the total cost. Moreover, for finite *r* of the highly utilized 400x2 case, DL8 has worse results than DL4.

200x2 and 400x3 cases are plotted in Figure 5.5 separately for finite and infinite *r*. For finite *r* cases, total cost decreases by 23% from DL1 to DL8 whereas the improvement from DL4 to DL8 with respect to DL1 is only 1%. For infinite *r* cases, total cost decreases by 29% from DL1 to DL8 whereas the improvement from DL4 to DL8 with respect to DL1 is 2%. Infinite *r* cases have higher improvement since no relocation cost is charged for SDFLP whereas increasing relocation cost by department disaggregation causes a smaller improvement for finite *r* cases. It is seen from Figure 5.5 (b) and (d) that ML1 and ML2 have similar patterns for total cost since total workload is the same in both situations.

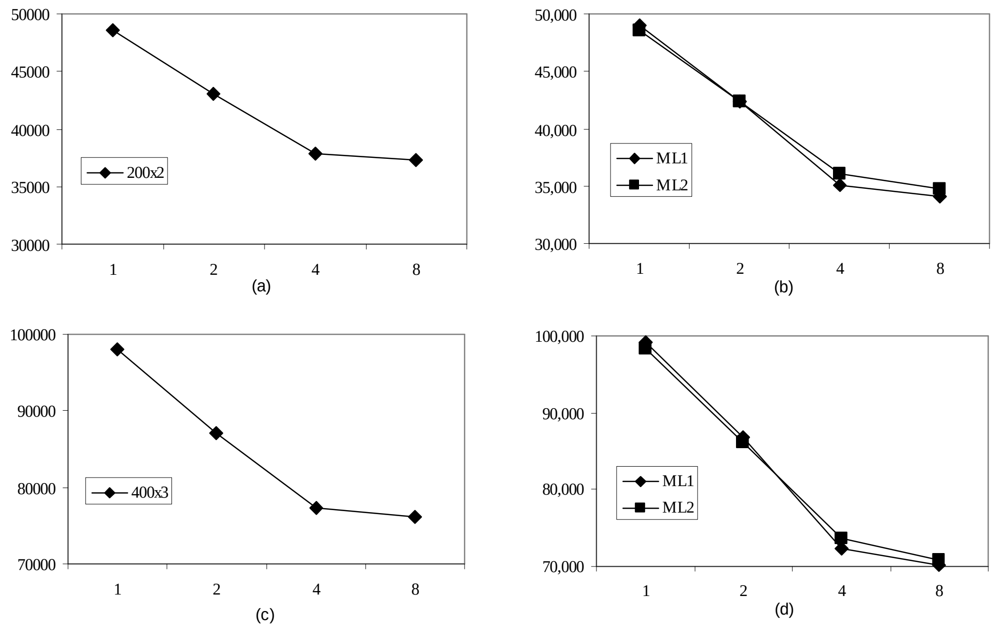


Figure 5.5. Total cost vs. duplication level, (a) 200x2 cases for finite *r*,
(b) 200x2 cases for infinite *r*, (c) 400x3 cases for finite *r*, (d) 400x3 cases for infinite *r*

In addition, according to the individual graphics in Figure E.2, 400x2 has higher cost than 400x3. The resulting configuration found by DSALAB for D = 400 is better simulated by 400x3 since it has shorter interperiod transition time than 400x2.

Work-In-Process (WIP)

We have considered WIP as the main measure of congestion in the system. Production lead times might also have been considered for this purpose. However, there are 8 product types each having a different processing route in each of the 30 problems, and their production lead times show such a large variability that the averages are not very meaningful.

Increasing the duplication level increases WIP in all cases of Figure E.3. Increase in WIP by disaggregation can be explained by the utilization of department duplicates. When the machines of a department are grouped to lay together in DL1, overutilization risk is smaller. All products are routed to a single department duplicate having 8 parallel machines, and the probability of finding an empty resource is high for the product coming to a department. On the other hand, for the case of DL8, there is only one machine in each department duplicate. Flow allocation aims at reducing the transportation cost, and the workload is not equally distributed among the duplicates. Some duplicates have more flow allocated to them, they are highly utilized, and they cause WIP accumulation and congestion.

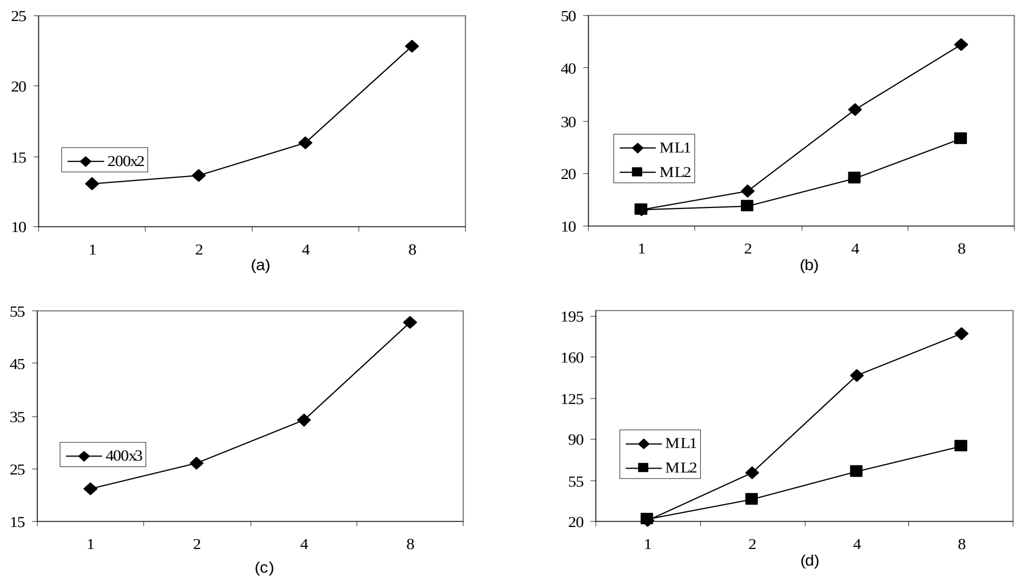


Figure 5.6. WIP vs. duplication level, (a) 200x2 cases for finite r,
(b) 200x2 cases for infinite r, (c) 400x3 cases for finite r, (d) 400x3 cases for infinite r

Specifically, 400x2 runs and $r = \infty$ case of 400x3 runs, have much longer interperiod transition times and close to 100% machine utilizations, resulting in very high WIP levels. This is an indication of a highly congested system. Actually, 400x2 case does not seem to have a sufficiently long period length. Therefore, 400x2 results are excluded from the overall average results shown in Figure 5.6 which are plotted for 200x2 and 400x3 cases and finite and infinite *r*. For the finite r cases in Figure 5.6 (a) and (c), increase up to DL4 is relatively smaller. Total increase from DL1 to DL8 is 112% whereas it is 42% from DL1 to DL4.

For the case of $r=\infty$ and 400x3 given in Figure 5.6 (d), the WIP levels are much higher even though a long period length is used. In Figure 5.6 (b) and (d), when $r=\infty$, no relocations are allowed. This case corresponds to the static distributed layout or SDFLP. The reason why WIP is higher in this case is that congestion in infinite *r* case is high at certain machines compared to finite *r* case. For infinite *r*, maximally used department duplicate utilization is 51%, 73% and 60% on the average for 200x2, 400x2 and 400x3, respectively. On the other hand, for finite r, maximally used department duplicate utilization is 43%, 70% and 52% for the same cases. Moreover, for DL1, the maximally used department duplicate utilization is nearly the same for all other factor combinations. It is independent of *r* for the finite case. However, for other duplication levels, it is larger for infinite *r* compared to finite *r*. For finite *r* cases, flow allocations are changed for each period during the optimization procedure, however SDFLP is a single period problem. In the mathematical model, capacities of department duplicates in SDFLP are used as eight times larger compared to finite *r* cases since eight periods are aggregated. Also, the flow allocation remains unchanged throughout all 8 periods whereas the demand changes. Therefore, unbalanced flow allocation risk is higher for SDFLP, resulting in high interperiod transition percentages, utilizations, WIP levels and congestion.

High WIP is more pronounced in demand mix level ML1, since workloads of some departments are high compared to others in ML1 whereas workloads of departments are balanced in later periods in ML2. Therefore, risk of overutilization is high in ML1 compared to ML2.

WIP versus Total cost

Individual graphics for the WIP versus the total cost and the total transporter utilization are given in Figures E4 and E5. Since travel cost is a linear function of the transporter utilization and relocation cost has a small share in total cost, Figures E.4 and E.5 have similar characteristics. Therefore, we concentrate on the relationship between total cost and WIP in this section. The highly utilized 400x2 case is excluded from the overall average results shown in Figure 5.7.

The mathematical model considers only the transportation and relocation costs. WIP is not considered in the optimization procedure and it is only observed in the simulation. Therefore, WIP is observed according to the minimum transportation and relocation costs configuration.

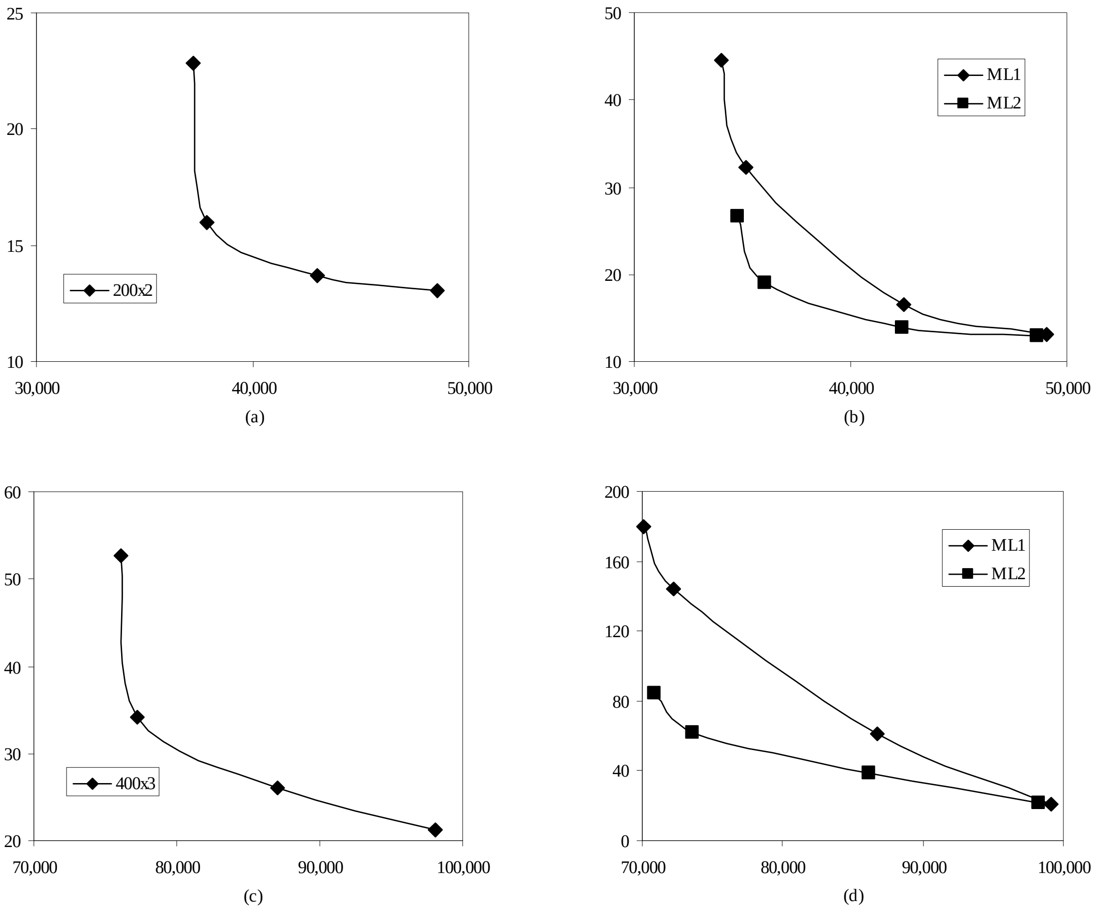


Figure 5.7. WIP (vertical axis) vs. total cost (horizontal axis), (a) 200x2 cases for finite *r*, (b) 200x2 cases for infinite *r*, (c) 400x3 cases for finite *r*, (d) 400x3 cases for infinite *r*

If we consider the total cost and WIP as two objectives, there is a clear tradeoff between the two, as shown in Figures E.4 and 5.7. According to Figure 5.7, increasing the duplication level reduces the transportation cost but increases the WIP. From DL4 to DL8, a decrease of

100 units in total cost increases WIP by 1.3, 0.7 and 1.4 unit loads for finite *r*, ML2 of infinite *r* and ML1 of infinite r, respectively. From DL1 to DL4, a decrease of 100 units in total cost increases WIP by 0.04, 0.1 and 0.3 unit loads for the same cases. Therefore, duplication from DL1 to DL4 has a smaller rate of increase of WIP compared to duplication from DL4 to DL8.

The overall rate of increase in WIP from DL1 to DL8 is 0.11, 0.16 and 0.38 for finite *r*, ML2 of infinite *r* and ML1 of infinite *r*, respectively. The rate is higher in infinite *r* cases, particularly in ML1 of infinite *r* case, since these cases are more congested compared to finite *r* and ML2 cases. The reason for the congestion in infinite *r* and ML1 cases are explained in the work-in-process section.

On the average, total cost decreases by 23% from DL1 to DL8 and 2% from DL4 to DL8. WIP, on the other hand, increases by 42% from DL1 to DL4 and 70% from DL4 to DL8 with respect to DL1, for finite *r* cases.

CHAPTER 6

CONCLUSION

Concluding Remarks

In this thesis, we have studied the facility layout problem and explored the effect of facility layout design on the operational performance of the production environment. Our study concerns three perspectives: the dynamic facility layout problem (DFLP), integration of empty travel cost into the layout problem formulation and dynamic distributed facility layout problem (DDFLP).

We focus on equal sized departments, which represent machine tools used in production. According to our literature review of the dynamic facility layout problem, simulated annealing (SA) is a promising solution method for DFLP. We propose a modification on SA II algorithm of McKendall et al. (2006) by slightly changing its neighborhood definition. Parameters of the new algorithm, SALAB, are experimentally set. We propose four variants of SALAB as SA*, SALAB-R, SALAB-1 and SALAB-2. After updating the best known solutions using the ones found in this study, SALAB-R finds the best solution for 31 of the 48 problems of Balakrishnan and Cheng (2000) and has the smallest average deviation, 0.05%, from the best known solutions. It is followed by SA II of McKendall et al. (2006) with 23 best results and an average deviation of 0.10%. 18 new best solutions are obtained in this study, 3 by SA*, 11 by SALAB-R, 2 by SALAB-1 and 4 by SALAB-2.

We then modify SALAB-R, renamed as DSALAB, to solve the dynamic distributed facility layout problem, which tries to minimize total (full and empty) travel cost and relocation cost. Several experimental factors are used to create dynamic environments. Finally, simulation is used to explore the operational performance of varying duplication levels and demand characteristics for the dynamic distributed layouts obtained by DSALAB.

We observe that DSALAB results do not deviate much from the simulation results except for the finite relocation cost runs of the high utilization case. The high utilization case has large interperiod transition times that cause a relatively large deviation. Infinite relocation cost case represents the static facility layout problem (SDFLP), in which decisions do not change over periods, resulting in small deviations although having relatively high utilization levels. According to improvement characteristics of DSALAB, full travel cost of the initial configuration is improved more compared to the empty travel cost. DSALAB improves the initial solution by 17% on the average.

Total cost decreases by disaggregation of departments into smaller ones, but the improvement is insignificant when departments have more than four duplicates. Moreover, it is observed that relocation cost has the smallest share in the total cost of the dynamic distributed layouts.

According to simulation results, by increasing the duplication level, total transporter utilization and total cost reduces whereas WIP increases. However, disaggregation into more than four department duplicates has a relatively insignificant improvement in total transporter utilization and total cost. The average reduction in total cost is 23% from DL1 to DL4 and 2% from DL4 to DL8. It is also observed that WIP has a smaller increase up to duplicate level four compared to that for disaggregation from duplicate level four to eight. For the finite *r* cases, the average increase in WIP is 42% from DL1 to DL4 and 70% from DL4 to DL8 with respect to DL1. According to the tradeoff curves of WIP and total cost, the tradeoff is stronger for the more utilized cases.

As a result, the fully distributed case, in which department duplicates have only one machine, is not necessarily the optimum when both the travel cost and the WIP are considered. Moreover, duplication level four (DL4) is a critical point for the problems and factors used in this study since further disaggregation does not significantly improve the travel and relocation costs whereas it largely increases the WIP. This is because overutilization risk of department duplicates is high in small sized duplicates having fewer parallel machines.

Future Research Issues

Proposed algorithm for DFLP, SALAB-R is a meta-heuristic implementation for the dynamic layout problem. Therefore it can be improved by hybrid applications. Other initial layout configurations and restart of the algorithm can also be considered. Proposed algorithm for DDFLP, DSALAB is also a meta-heuristic and can be improved by hybrid applications.

Moreover some other solution methods can be studied for the DDFLP model which is a nonlinear mixed integer problem. In particular, the empty travel cost can be considered in flow allocation in addition to the full travel cost.

For the DDFLP model, we assume that department duplicates have equal capacities to have a balanced utilization and less congested system. However effects of other strategies and slack capacities on material handling cost and WIP can be further analyzed.

We assume that relocation cost of duplicates is same for all departments but it can vary over department types. In relocation, there is also the opportunity cost of not being able to produce anything. This can be reflected in the formulation by making the relocated machine unavailable for one period.

We introduce the empty travel cost of transporter under the assumption that there is one transporter or random selection rule from multiple transporters is implemented. However other strategies can be formulated and introduced in dynamic distributed layout formulations. Better strategies for different performance measures can be found in this manner.

The overall demand volume and overall available area can be made dynamic as well. One may increase demand volume and add new machines over time, implying expansion of the facility.

Another factor affecting the performance of the production environment is the production and transportation batch sizes of products. In this study, we assume that production and transportation batches have the same size (unit load). Moreover no set-up time is incurred while produced part type is changed. Mathematical formulation and simulation model can be modified to include these factors.

We analyze the distributed facility considering equal sized duplicates which equally share the machines of the same department type. However duplicates with varying sizes can be considered for the distributed layout design. The layout problem can be studied on continuous scale instead of using the grid layout.

We consider the center of the rectangular area as the input and output point of a department duplicate. A more realistic approach can be applied by using different input and output points and deciding on their locations around the area of department duplicate. Location of entrance and exit points to and from the system may also be considered as additional decisions.

Distributed layout may have advantages on production lead times since it has the flexibility to construct alternative short processing paths considering the processing routes of products. Therefore, one may concentrate on distributed layout models that consider lead times or WIP. We use the relocation cost and full and empty travel costs of transporter as the objective of layout design. However, an approximate closed form WIP or lead time formulation can be developed using queuing network results and used in the objective function. Moreover, a multi-criteria approach can be used in both the dynamic and distributed layout design problems, considering various cost components and operational performance measures.

REFERENCES

Azadivar, F. and Wang J.J., 2000. Facility layout optimization using simulation and genetic algorithms *International Journal of Production Research* **38** (17), 4369-4383.

Balakrishnan, J. and Cheng, C.H., 1998. Dynamic layout algorithms: a State-of-the-art Survey *Omega, International Journal of Management Science* **26** (4), 507-521.

Balakrishnan, J. and Cheng, C.H., 2000. Genetic search and the dynamic layout problem *Computers & Operations Research* **27**, 587-593.

Balakrishnan, J. and Cheng, C.H., 2006. A note on “a hybrid genetic algorithm for the dynamic plant layout problem” *International Journal of Production Economics* **103**, 87-89.

Balakrishnan, J., Cheng, C.H. and Conway, D.G., 2000. An improved pair-wise exchange heuristic for the dynamic plant layout problem *International Journal of Production Research* **38** (13), 3067-3077.

Balakrishnan, J., Cheng, C.H., Conway, D.G. and Lau, C.M., 2003. A hybrid genetic algorithm for the dynamic plant layout problem *International Journal of Production Economics* **86**, 107-120.

Balakrishnan, J., Jacobs, F.R. and Venkataramanan, M.A., 1992. Solutions for the constrained dynamic facility layout problem *European Journal of Operational Research* **57**, 280-286.

Balakrishnan, J., 1993. Notes: “The Dynamics of Plant Layout” *Management Science* **39** (5), 654-655.

Batta, R., 1987. Comment on “The Dynamics of Plant Layout” *Management Science* **33** (8), 1065.

Baykasoğlu, A. and Gindy, N.N.Z., 2001. A simulated annealing algorithm for dynamic layout problem *Computers & Operations Research* **28**, 1403-1426.

Baykasoğlu, A. and Gindy, N.N.Z., 2004. Erratum to “A simulated annealing algorithm for dynamic layout problem” *Computers & Operations Research* **31**, 313-315.

Baykasoğlu, A., Dereli T. and Sabuncu, I., 2006. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems *Omega, The International Journal of Management Science* **34**, 385-396.

Benjaafar, S. and Sheikhzadeh, M., 2000. Design of flexible plant layouts *IIE Transactions* **32**, 309-322.

Benjaafar, S., Heragu, S.S. and Irani, S.A., 2002. Next Generation Factory Layouts: Research Challenges and Recent Progress *Interfaces* **32** (6), 58-76.

Benjaafar, S., 2002. Modeling and Analysis of Congestion in the Design of Facility Layouts *Management Science* **48** (5), 679-704.

Castillo, I. and Peters B.A., 2002. Unit load and material-handling considerations in facility layout design *International Journal of Production Research* **40** (13), 2955-2989.

Castillo, I. and Peters B.A., 2003. An extended distance-based facility layout problem *International Journal of Production Research* **41** (11), 2451-2479.

Chang, M., Ohkura, K., Ueda, K. and Sugiyama, M., 2002. A symbiotic evolutionary algorithm for dynamic facility layout problem *Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02)* **2**, 1745-1750.

Conway, D.G. and Venkataramanan M.A., 1994. Genetic search and the dynamic facility layout problem *Computers & Operations Research* **21** (8), 955-960.

Cury, G.L., Peters, B.A. and Lee, M., 2003. Queueing network model for a class of material-handling systems *International Journal of Production Research* **41** (16), 3901-3920.

Drira, A., Pierreval, H. and Hajri-Gabouj, S., 2007. Facility layout problems: A survey *Annual Reviews in Control* **31**, 255-267.

Dowsland, K.A., 1995. Simulated Annealing *CHAPTER 2 in Modern Heuristic Techniques for Combinatorial Problems* (Edited by C.R. Reeves), McGraw-Hill, UK.

Erel, E., Ghosh, J.B. and Simon J.T., 2005. Corrigendum New heuristic for the dynamic layout problem *Journal of the Operational Research Society* **56**, 1001.

Erel, E., Ghosh, J.B. and Simon J.T., 2003. New heuristic for the dynamic layout problem *Journal of the Operational Research Society* **54**, 1275-1282.

Fu, M.C. and Kaku, B.K., 1994. Minimizing Work-in-Process and Material Handling in the Facilities Layout Problem *Technical Research Report T.R. 95-41*, the University of Maryland, College Park, Institute for Systems Research.

Fu, M.C. and Kaku, B.K., 1997. Minimizing work-in-process and material handling in the facilities layout problem *IIE Transactions* **29**, 29-36.

Johnson, M.E., 2001. Modeling empty vehicle traffic in AGVS design *International Journal of Production Research* **39** (12), 2615-2633.

Kaku, B.K. and Mazzola, J.B., 1997. A Tabu-Search Heuristic for the Dynamic Plant Layout Problem *INFORMS Journal on Computing* **9** (4), 374-384.

Kouvelis, P. and Kiran, A.S., 1990. The plant layout problem in automated manufacturing systems *Annals of Operations Research* **26**, 397-412.

Kulturel-Konak, S., 2007. Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century *Journal of Intelligent Manufacturing* **18**, 273-284.

Lacksonen, T.A. and Enscore E.E., 1993. Quadratic assignment algorithms for the dynamic layout problem *International Journal of Production Research* **31** (3), 503-517.

Lahmar, M. and Benjaafar, S., 2005. Design of distributed layouts *IIE Transactions* **37**, 303-318.

McKendall, A.R. and Shang J., 2006. Hybrid ant systems for the dynamic facility layout problem *Computers & Operations Research* **33**, 790-803.

McKendall, A.R., Shang, J. and Kuppusamy, S., 2006. Simulated annealing heuristics for the dynamic facility layout problem *Computers & Operations Research* **33**, 2431-2444.

Rosenblatt, M.J., 1986. The dynamics of plant layout *Management Science* **32** (1), 76-86.

Urban, T.L., 1993. A heuristic for the dynamic facility layout problem *IIE Transactions* **25** (4), 57-63.

Urban, T.L., 1998. Solution procedures for the dynamic facility layout problem *Annals of Operations Research* **76**, 323-342.

Urban, T.L., Chiang, W.C. and Russell, R.A., 2000. The integrated machine allocation and layout problem *International Journal of Production Research* **38** (13), 2911-2930.

APPENDIX A

EXPERIMENTS FOR ITERATION LIMIT AND COOLING RATIO SETTINGS

6D5P

A: 30, 180, 360, 720, 1440, 1500, 1600, 2000
α: 0.95, 0.99, 0.996, 0.997, 0.998
Each A and α combination is tested on 5 problems with 3 different initial solutions

6D10P

A:360, 720, 1440, 2400, 2500, 2880, 3000
α:0.99, 0.996, 0.997, 0.998
Each A and α combination is tested on 5 problems with 3 different initial solutions

15D5P

A:75, 1125, 2000, 2500, 3000, 3200, 3800
α:0.99, 0.996, 0.997, 0.998
Each A and α combination is tested on 5 problems with 3 different initial solutions

15D10P

A:2250, 5000, 6000, 6500, 7000, 7500
α:0.996, 0.997, 0.998
Each A and α combination is tested on 3 problems with 3 different initial solutions

30D5P

A:6000, 7000, 7800, 8500, 10000
α: 0.996, 0.997, 0.998
Each A and α combination is tested on 3 problems with 3 different initial solutions

30D10P

A:10000, 12000, 15000
α:0.996, 0.997, 0.998
Each A and α combination is tested on 3 problems with 3 different initial solutions

The following notation is used for reporting the results in Figures A.1 through A.6 and Tables A.1 through A.6:

Alpha: Cooling ratio
A: Number of moves permitted at each temperature level, limit on inner loop iteration count.
Last Best N: Number of moves until the best solution is found (average of the problems tested at each α and A combination).
Dev % : Percentage deviation of the best cost found with SALAB from the best known cost reported by McKendall et al. (2006) (average of the problems tested at each α and A combination).
Best: Number of times the best known solution is found for problems tested at each α and A combination.

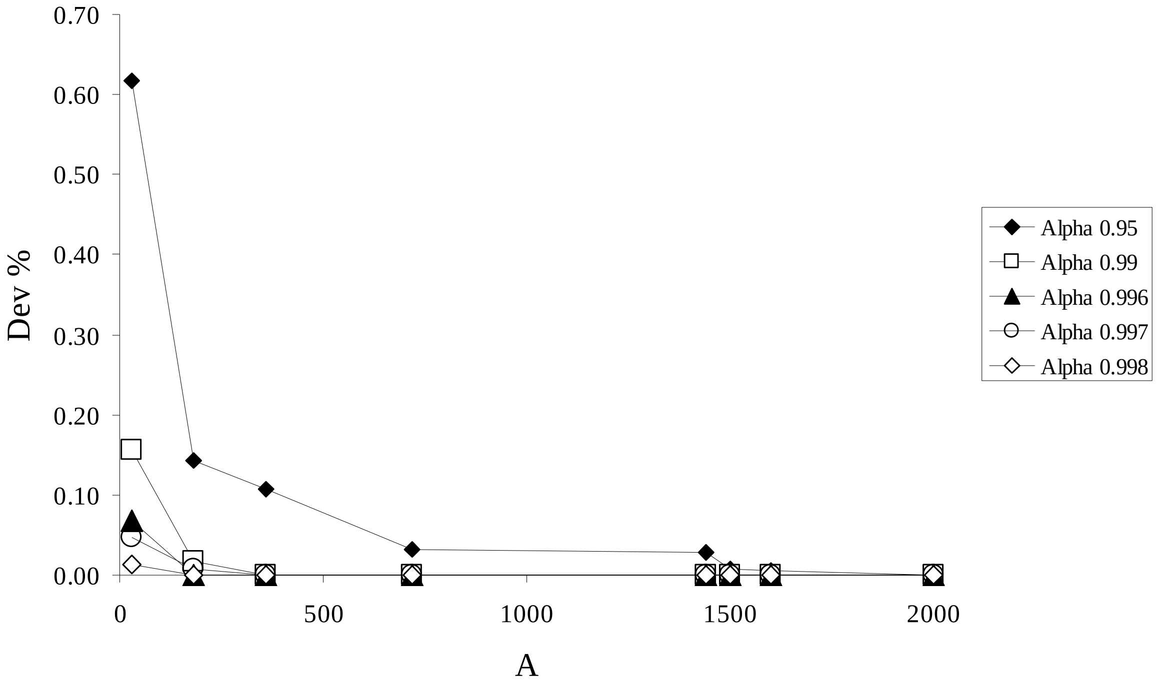


Figure A.1. Dev % results for 6D5P settings

Table A.1. 6D5P Settings

Alpha	A	Last Best N	Dev %	Best
0.990	360	68,116	0.00	15
0.950	2,000	74,294	0.00	15
0.996	180	88,076	0.00	15
0.990	720	128,563	0.00	15
0.998	180	130,454	0.00	15
0.996	360	141,324	0.00	15
0.997	360	200,683	0.00	15
0.990	1,440	219,662	0.00	15
0.998	360	237,067	0.00	15
0.990	1,600	237,602	0.00	15
0.990	1,500	249,269	0.00	15
0.996	720	260,893	0.00	15
0.990	2,000	295,958	0.00	15
0.997	720	329,703	0.00	15
0.996	1,440	379,722	0.00	15
0.996	1,500	433,793	0.00	15
0.998	720	446,453	0.00	15
0.996	1,600	464,548	0.00	15
0.997	1,440	534,800	0.00	15
0.996	2,000	576,999	0.00	15
0.997	1,500	613,737	0.00	15
0.997	2,000	627,669	0.00	15
0.997	1,600	636,856	0.00	15
0.998	1,440	700,554	0.00	15
0.998	1,500	707,356	0.00	15
0.998	1,600	708,910	0.00	15
0.998	2,000	831,996	0.00	15
0.950	1,600	68,120	0.00	13
0.950	1,500	59,097	0.01	14
0.997	180	111,949	0.01	14
0.998	30	34,878	0.01	13
0.990	180	37,245	0.02	12
0.950	1,440	56,229	0.03	14
0.950	720	29,524	0.03	10
0.997	30	20,808	0.05	11
0.996	30	18,190	0.07	11
0.950	360	15,169	0.11	8
0.950	180	7,068	0.14	8
0.990	30	8,160	0.16	7
0.950	30	1,918	0.62	3

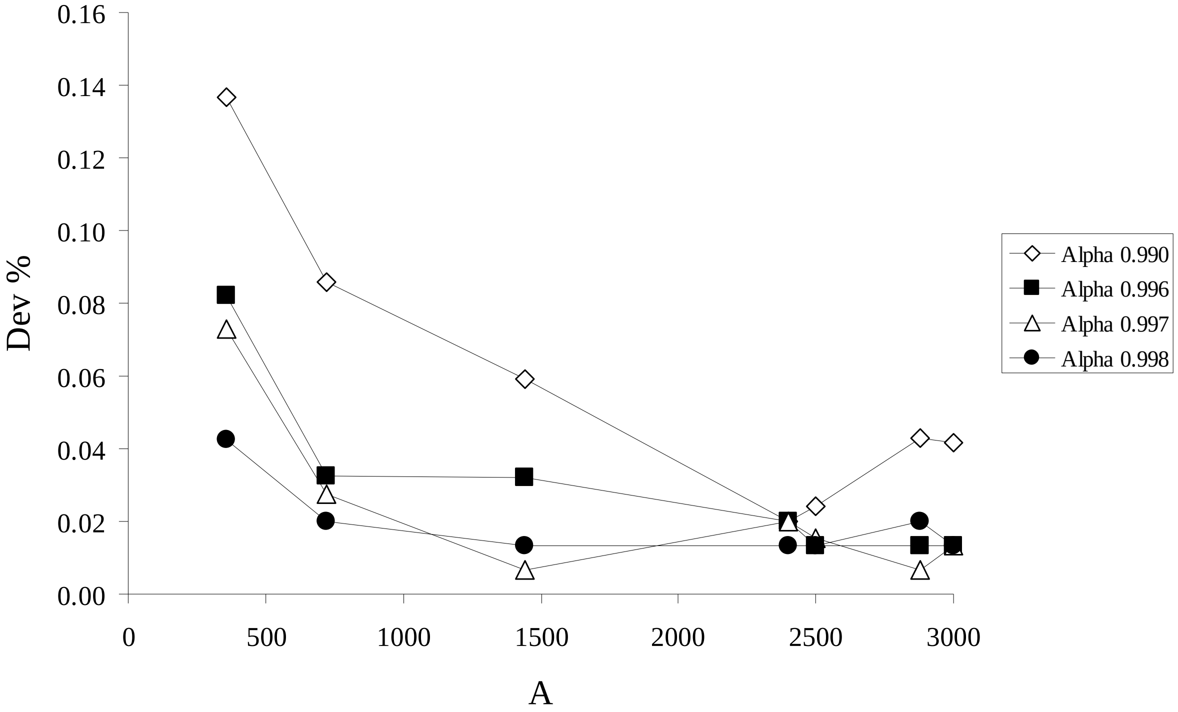


Figure A.2. Dev % results for 6D10P settings

Table A.2. 6D10P Settings

Alpha	A	Last Best N	Dev %	Best
0.997	1,440	1,627,259	0.01	14
0.997	2,880	3,106,051	0.01	14
0.996	2,500	2,020,057	0.01	13
0.998	1,440	2,275,422	0.01	13
0.996	2,880	2,384,307	0.01	13
0.996	3,000	2,426,053	0.01	13
0.997	3,000	3,090,197	0.01	13
0.998	2,400	3,626,606	0.01	13
0.998	2,500	3,820,747	0.01	13
0.998	3,000	4,472,275	0.01	13
0.997	2,500	2,625,237	0.02	12
0.990	2,400	825,817	0.02	12
0.998	720	1,195,652	0.02	12
0.996	2,400	1,855,140	0.02	12
0.997	2,400	2,392,610	0.02	12
0.998	2,880	4,255,731	0.02	12
0.990	2,500	824,887	0.02	12
0.997	720	812,376	0.03	11
0.996	1,440	1,117,920	0.03	9
0.996	720	609,453	0.03	9
0.990	3,000	990,425	0.04	12
0.998	360	616,978	0.04	9
0.990	2,880	943,265	0.04	9
0.990	1,440	510,563	0.06	7
0.997	360	425,041	0.07	7
0.996	360	314,799	0.08	5
0.990	720	264,107	0.09	6
0.990	360	129,850	0.14	5

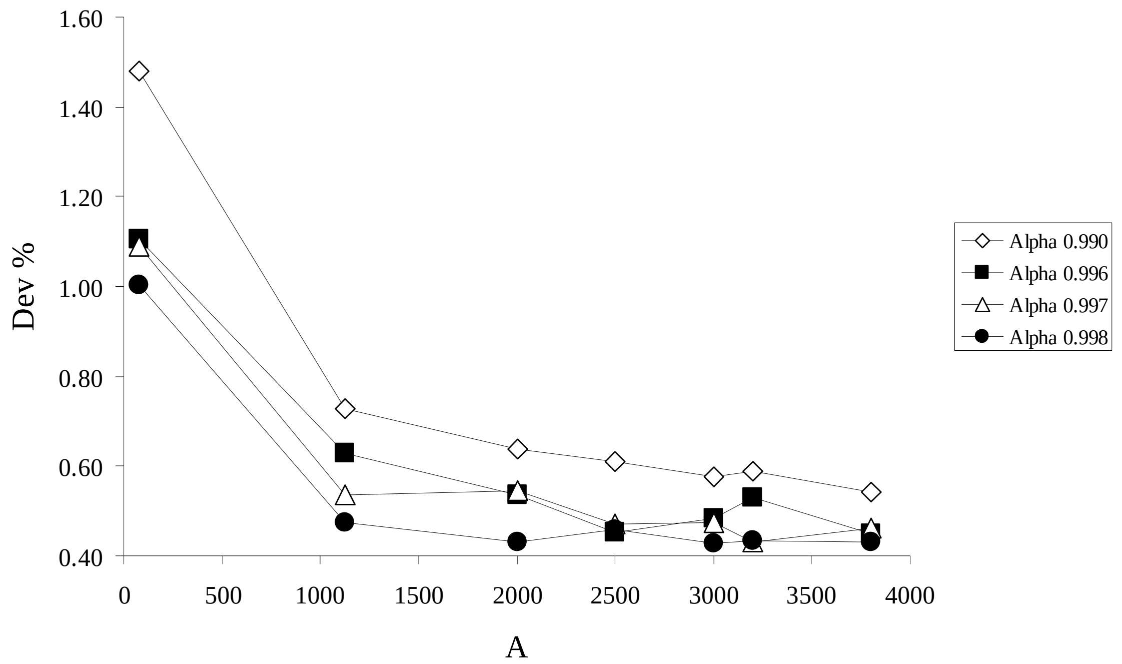


Figure A.3. Dev % results for 15D5P settings

Table A.3. 15D5P Settings

Alpha	A	Last Best N	Dev %	Best
0.998	3,000	7,636,484	0.43	1
0.998	2,000	5,002,198	0.43	2
0.998	3,800	9,382,435	0.43	1
0.997	3,200	5,303,326	0.43	2
0.998	3,200	7,764,491	0.43	2
0.996	3,800	4,838,619	0.45	3
0.996	2,500	3,181,836	0.45	1
0.998	2,500	6,376,310	0.46	1
0.997	3,800	6,249,410	0.46	2
0.997	2,500	4,111,355	0.47	1
0.997	3,000	5,035,647	0.47	2
0.998	1,125	2,811,060	0.48	2
0.996	3,000	3,764,917	0.48	2
0.996	3,200	4,060,723	0.53	0
0.997	1,125	1,923,593	0.53	0
0.996	2,000	2,543,759	0.54	1
0.990	3,800	1,936,138	0.54	1
0.997	2,000	3,456,693	0.54	0
0.990	3,000	1,519,597	0.58	0
0.990	3,200	1,618,461	0.59	0
0.990	2,500	1,256,665	0.61	0
0.996	1,125	1,433,662	0.63	0
0.990	2,000	1,067,050	0.64	0
0.990	1,125	583,175	0.73	0
0.998	75	205,300	1.00	0
0.997	75	136,860	1.09	0
0.996	75	105,533	1.11	0
0.990	75	43,333	1.48	0

Table A.4. 15D10P Settings

Alpha	A	Last Best N	Dev %	Best
0.998	7,500	21,811,182	0.26	0
0.998	6,000	17,575,041	0.27	1
0.997	7,500	14,652,396	0.28	0
0.996	7,000	10,558,336	0.29	2
0.998	5,000	14,644,984	0.29	1
0.997	6,000	11,781,145	0.29	0
0.997	6,500	12,575,895	0.29	2
0.997	5,000	9,763,056	0.35	0
0.998	6,500	19,679,266	0.36	0
0.996	5,000	7,492,534	0.36	0
0.996	6,500	10,021,095	0.37	1
0.997	7,000	14,085,384	0.37	0
0.998	7,000	20,322,481	0.38	0
0.996	7,500	11,191,619	0.39	0
0.996	6,000	8,834,196	0.40	0
0.998	2,250	6,773,309	0.42	0
0.996	2,250	3,404,277	0.49	0
0.997	2,250	4,578,746	0.51	0

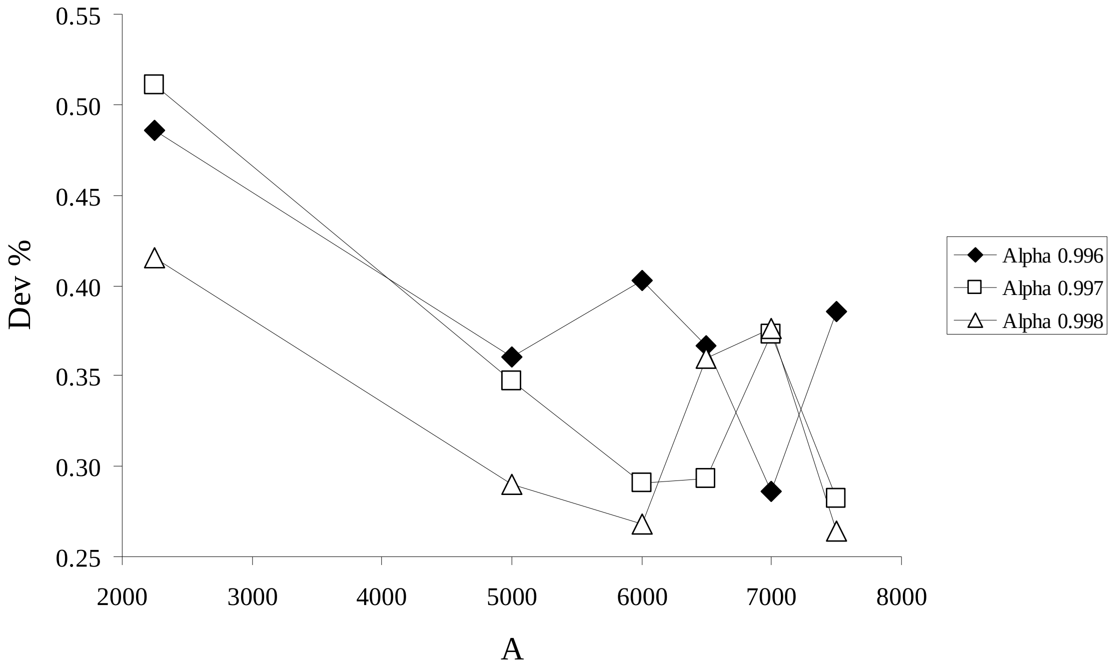


Figure A.4. Dev % results for15D10P settings

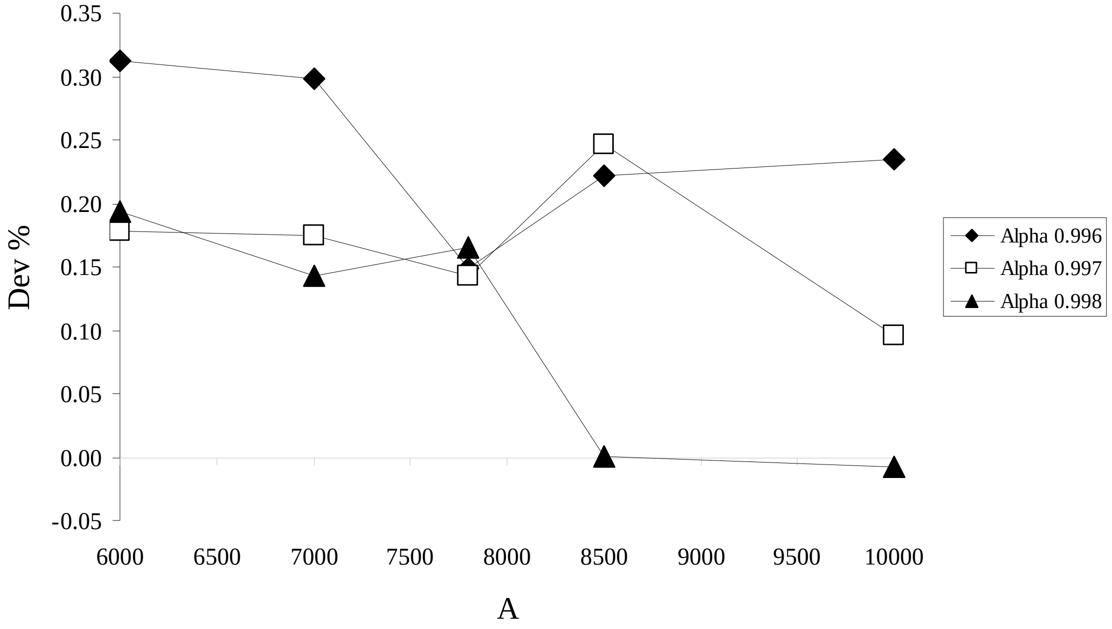


Figure A.5. Dev % results for 30D5P settings

Table A.5. 30D5P Settings

Alpha	A	Last Best N	Dev %	Best
0.998	10,000	28,970,108	-0.01	4
0.998	8,500	25,137,849	0.00	6
0.997	10,000	19,638,710	0.10	3
0.997	7,800	15,444,749	0.14	2
0.998	7,000	20,435,752	0.14	3
0.996	7,800	11,721,288	0.15	3
0.998	7,800	22,892,674	0.17	2
0.997	7,000	14,059,368	0.18	1
0.997	6,000	11,683,826	0.18	3
0.998	6,000	17,797,077	0.19	2
0.996	8,500	12,666,771	0.22	3
0.996	10,000	14,934,010	0.23	1
0.997	8,500	16,517,165	0.25	1
0.996	7,000	9,989,832	0.30	0
0.996	6,000	8,886,193	0.31	1

Table A.6. 30D10P Settings

Alpha	A	Last Best N	Dev %	Best
0.998	12,000	40,295,472	0.10	5
0.998	15,000	49,867,332	0.11	5
0.998	10,000	34,600,373	0.14	4
0.997	12,000	25,922,308	0.19	4
0.997	15,000	33,136,543	0.19	3
0.997	10,000	22,017,694	0.21	3
0.996	10,000	16,310,018	0.21	3
0.996	12,000	20,475,555	0.23	3
0.996	15,000	25,230,101	0.30	2

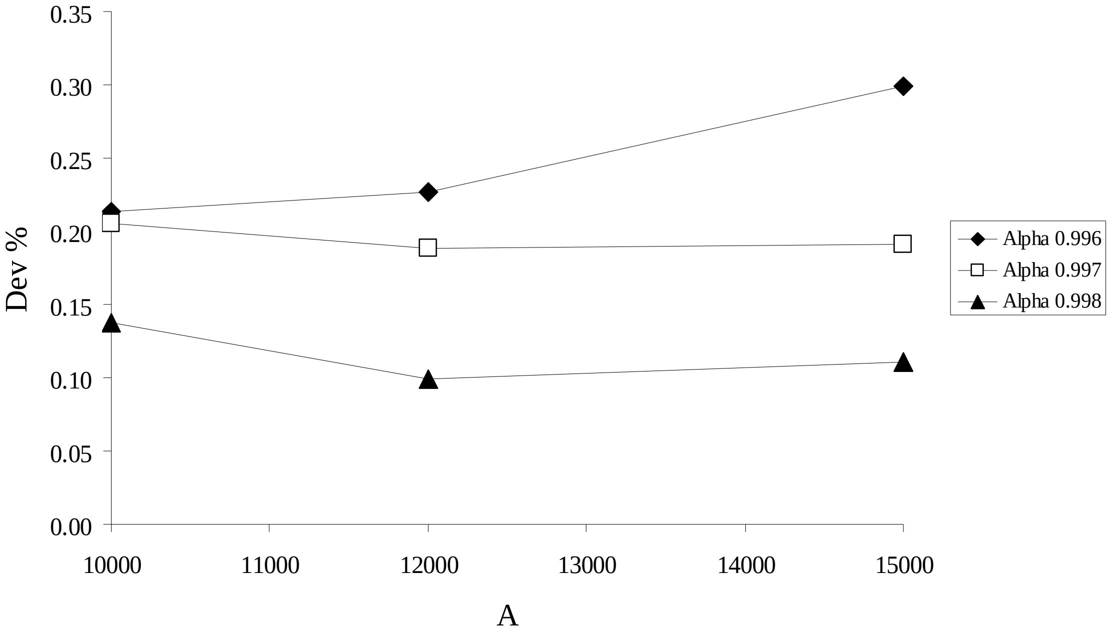


Figure A.6. Dev % results for 30D10P settings

APPENDIX B

DETAILED RESULTS FOR SALAB

SA*, SALAB-R, SALAB-1 and SALAB-2 are tested on 48 benchmark problems of Balakrishnan and Cheng (2000).

For SA* and SALAB-R, each problem is solved five times starting with different random initial solutions. Both the average and minimum of these five results are reported in Tables B.1 through B.4. Notation used in these tables is as follows.

BESTKNOWN: Best known solution before SALAB runs

PR#: Problem number

D: Number of departments

P: Number of Periods

AVGCOST: Average of five for SA* and SALAB-R

ADEV%: Percentage deviation of AVGCOST from BESTKNOWN for SA* and SALAB-R

MINCOST: Best of 5 for SA* and SALAB-R

MDEV%: Percentage deviation of MINCOST from BESTKNOWN for SA* and SALAB-R

DEV%: Percentage deviation of COST from BESTKNOWN from SALAB-1 and SALAB-2

CPUMIN: Average CPU time in minutes on a Pentium IV 1.8GHz PC.

AVGDEV: Average percentage deviation over all 48 problems

Table B.1. SA* results

Pr#	D	P	AVG COST	ADEV%	MIN COST	MDEV%	BESTKNOWN	CPU MIN
1	6	5	106,419	0.00	106,419	0.00	106,419	0.20
2	6	5	104,834	0.00	104,834	0.00	104,834	0.20
3	6	5	104,320	0.00	104,320	0.00	104,320	0.20
4	6	5	106,399	0.00	106,399	0.00	106,399	0.20
5	6	5	105,628	0.00	105,628	0.00	105,628	0.20
6	6	5	103,985	0.00	103,985	0.00	103,985	0.20
7	6	5	106,439	0.00	106,439	0.00	106,439	0.20
8	6	5	103,771	0.00	103,771	0.00	103,771	0.20
9	6	10	214,313	0.00	214,313	0.00	214,313	0.38
10	6	10	212,175	0.02	212,134	0.00	212,134	0.38
11	6	10	208,174	0.09	207,987	0.00	207,987	0.38
12	6	10	212,742	0.10	212,741	0.10	212,530	0.38
13	6	10	210,906	0.00	210,906	0.00	210,906	0.38
14	6	10	209,952	0.01	209,932	0.00	209,932	0.38
15	6	10	214,252	0.00	214,252	0.00	214,252	0.38
16	6	10	212,588	0.00	212,588	0.00	212,588	0.38
17	15	5	481,069	0.13	480,453	0.00	480,453	0.96
18	15	5	485,584	0.17	484,761	0.00	484,761	0.96
19	15	5	490,114	0.28	489,059	0.06	488,748	0.96
20	15	5	486,379	0.41	484,446	0.01	484,405	0.96
21	15	5	488,842	0.23	488,044	0.07	487,722	0.96
22	15	5	487,910	0.25	486,898	0.04	486,685	0.96
23	15	5	487,471	0.14	486,855	0.02	486,779	0.97
24	15	5	492,109	0.26	491,589	0.16	490,812	0.96
25	15	10	980,916	0.15	978,848	-0.06	979,468	2.96
26	15	10	979,062	0.10	978,324	0.03	978,065	2.96
27	15	10	984,298	0.64	983,007	0.51	978,027	2.96
28	15	10	974,807	0.21	972,432	-0.04	972,797	2.96
29	15	10	978,067	0.09	976,811	-0.04	977,188	2.96
30	15	10	971,148	0.36	969,443	0.19	967,617	2.96
31	15	10	981,063	0.20	979,166	0.01	979,114	2.95
32	15	10	984,907	0.13	983,486	-0.02	983,672	2.96
33	30	5	576,526	0.08	575,288	-0.13	576,039	7.63
34	30	5	570,483	0.42	569,403	0.23	568,095	7.63
35	30	5	574,197	0.08	573,214	-0.09	573,739	7.63
36	30	5	567,555	0.23	566,655	0.07	566,248	7.62
37	30	5	559,193	0.15	557,761	-0.11	558,353	7.62
38	30	5	567,506	0.25	566,715	0.11	566,077	7.63
39	30	5	568,891	0.31	567,230	0.02	567,131	7.63
40	30	5	576,396	0.46	575,345	0.28	573,755	7.63
41	30	10	1,163,451	0.02	1,160,830	-0.21	1,163,222	9.13
42	30	10	1,165,135	0.31	1,162,119	0.05	1,161,521	9.13
43	30	10	1,159,591	0.23	1,157,832	0.08	1,156,918	9.13
44	30	10	1,146,595	0.06	1,145,004	-0.08	1,145,918	9.16
45	30	10	1,153,421	2.40	1,129,705	0.29	1,126,432	9.17
46	30	10	1,155,110	1.21	1,144,154	0.25	1,141,344	9.17
47	30	10	1,150,933	0.89	1,148,459	0.68	1,140,744	9.17
48	30	10	1,168,020	0.57	1,166,727	0.46	1,161,437	9.17
			AVG DEV	0.24	AVG DEV	0.06		

Table B.2. SALAB-R Results

Pr#	D	P	AVG COST	ADEV%	MIN COST	MDEV%	BESTKNOWN	CPU MIN
1	6	5	106,419	0.00	106,419	0.00	106,419	0.15
2	6	5	104,834	0.00	104,834	0.00	104,834	0.15
3	6	5	104,320	0.00	104,320	0.00	104,320	0.15
4	6	5	106,399	0.00	106,399	0.00	106,399	0.15
5	6	5	105,628	0.00	105,628	0.00	105,628	0.15
6	6	5	103,985	0.00	103,985	0.00	103,985	0.15
7	6	5	106,439	0.00	106,439	0.00	106,439	0.15
8	6	5	103,771	0.00	103,771	0.00	103,771	0.15
9	6	10	214,313	0.00	214,313	0.00	214,313	0.26
10	6	10	212,134	0.00	212,134	0.00	212,134	0.26
11	6	10	207,987	0.00	207,987	0.00	207,987	0.27
12	6	10	212,572	0.02	212,530	0.00	212,530	0.26
13	6	10	210,906	0.00	210,906	0.00	210,906	0.27
14	6	10	209,944	0.01	209,932	0.00	209,932	0.27
15	6	10	214,252	0.00	214,252	0.00	214,252	0.26
16	6	10	212,588	0.00	212,588	0.00	212,588	0.26
17	15	5	480,696	0.05	480,453	0.00	480,453	0.79
18	15	5	484,801	0.01	484,761	0.00	484,761	0.79
19	15	5	489,479	0.15	488,748	0.00	488,748	0.79
20	15	5	484,477	0.01	484,446	0.01	484,405	0.79
21	15	5	488,230	0.10	487,822	0.02	487,722	0.79
22	15	5	487,272	0.12	486,493	-0.04	486,685	0.79
23	15	5	487,269	0.10	486,819	0.01	486,779	0.79
24	15	5	491,664	0.17	490,812	0.00	490,812	0.79
25	15	10	979,968	0.05	978,848	-0.06	979,468	2.23
26	15	10	979,318	0.13	978,523	0.05	978,065	2.22
27	15	10	983,273	0.54	981,191	0.32	978,027	2.23
28	15	10	973,747	0.10	971,740	-0.11	972,797	2.22
29	15	10	978,739	0.16	977,856	0.07	977,188	2.23
30	15	10	969,583	0.20	968,466	0.09	967,617	2.23
31	15	10	979,735	0.06	978,946	-0.02	979,114	2.23
32	15	10	984,477	0.08	984,023	0.04	983,672	2.21
33	30	5	575,570	-0.08	575,028	-0.18	576,039	6.43
34	30	5	569,252	0.20	568,057	-0.01	568,095	6.43
35	30	5	573,345	-0.07	572,478	-0.22	573,739	6.44
36	30	5	567,809	0.28	566,977	0.13	566,248	6.45
37	30	5	557,850	-0.09	556,269	-0.37	558,353	6.46
38	30	5	566,250	0.03	565,004	-0.19	566,077	6.45
39	30	5	568,736	0.28	568,097	0.17	567,131	6.43
40	30	5	575,109	0.24	573,193	-0.10	573,755	6.45
41	30	10	1,162,284	-0.08	1,160,949	-0.20	1,163,222	7.61
42	30	10	1,161,028	-0.04	1,159,629	-0.16	1,161,521	7.61
43	30	10	1,156,260	-0.06	1,153,389	-0.31	1,156,918	7.62
44	30	10	1,144,701	-0.11	1,143,000	-0.25	1,145,918	7.63
45	30	10	1,150,121	2.10	1,125,260	-0.10	1,126,432	7.72
46	30	10	1,153,148	1.03	1,142,579	0.11	1,141,344	7.64
47	30	10	1,147,241	0.57	1,144,875	0.36	1,140,744	7.64
48	30	10	1,166,409	0.43	1,164,198	0.24	1,161,437	7.65
			AVG DEV	0.14	AVG DEV	-0.01		

Table B.3. SALAB-1 results

Pr #	D	P	COST	DEV%	BESTKNOWN	CPU MIN
1	6	5	106,419	0.00	106,419	0.15
2	6	5	104,834	0.00	104,834	0.15
3	6	5	104,320	0.00	104,320	0.15
4	6	5	106,399	0.00	106,399	0.15
5	6	5	105,628	0.00	105,628	0.16
6	6	5	103,985	0.00	103,985	0.15
7	6	5	106,439	0.00	106,439	0.15
8	6	5	103,771	0.00	103,771	0.15
9	6	10	214,313	0.00	214,313	0.27
10	6	10	212,134	0.00	212,134	0.27
11	6	10	207,987	0.00	207,987	0.27
12	6	10	212,530	0.00	212,530	0.26
13	6	10	210,906	0.00	210,906	0.27
14	6	10	209,932	0.00	209,932	0.27
15	6	10	214,252	0.00	214,252	0.27
16	6	10	212,588	0.00	212,588	0.26
17	15	5	480,496	0.01	480,453	0.79
18	15	5	484,761	0.00	484,761	0.79
19	15	5	489,430	0.14	488,748	0.80
20	15	5	484,446	0.01	484,405	0.80
21	15	5	488,054	0.07	487,722	0.80
22	15	5	486,493	-0.04	486,685	0.80
23	15	5	487,698	0.19	486,779	0.80
24	15	5	491,237	0.09	490,812	0.79
25	15	10	979,977	0.05	979,468	2.26
26	15	10	979,002	0.10	978,065	2.26
27	15	10	981,528	0.36	978,027	2.26
28	15	10	974,389	0.16	972,797	2.26
29	15	10	978,876	0.17	977,188	2.27
30	15	10	971,987	0.45	967,617	2.27
31	15	10	981,114	0.20	979,114	2.26
32	15	10	984,865	0.12	983,672	2.26
33	30	5	575,813	-0.04	576,039	6.46
34	30	5	568,950	0.15	568,095	6.46
35	30	5	573,414	-0.06	573,739	6.46
36	30	5	566,387	0.02	566,248	6.48
37	30	5	557,657	-0.12	558,353	6.48
38	30	5	566,761	0.12	566,077	6.48
39	30	5	567,859	0.13	567,131	6.46
40	30	5	575,326	0.27	573,755	6.47
41	30	10	1,159,188	-0.35	1,163,222	7.67
42	30	10	1,160,299	-0.11	1,161,521	7.65
43	30	10	1,156,093	-0.07	1,156,918	7.66
44	30	10	1,144,305	-0.14	1,145,918	7.67
45	30	10	1,125,039	-0.12	1,126,432	7.67
46	30	10	1,144,254	0.25	1,141,344	7.66
47	30	10	1,147,985	0.63	1,140,744	7.67
48	30	10	1,160,420	-0.09	1,161,437	7.68
			AVG DEV	0.05		

Table B.4. SALAB-2 results

Pr #	D	P	COST	DEV%	BESTKNOWN	CPU MIN
1	6	5	106,419	0.00	106,419	0.15
2	6	5	104,834	0.00	104,834	0.15
3	6	5	104,320	0.00	104,320	0.15
4	6	5	106,399	0.00	106,399	0.15
5	6	5	105,628	0.00	105,628	0.15
6	6	5	103,985	0.00	103,985	0.15
7	6	5	106,439	0.00	106,439	0.15
8	6	5	103,771	0.00	103,771	0.15
9	6	10	214,313	0.00	214,313	0.27
10	6	10	212,134	0.00	212,134	0.26
11	6	10	207,987	0.00	207,987	0.27
12	6	10	212,741	0.10	212,530	0.26
13	6	10	210,906	0.00	210,906	0.27
14	6	10	209,932	0.00	209,932	0.27
15	6	10	214,252	0.00	214,252	0.27
16	6	10	212,588	0.00	212,588	0.26
17	15	5	480,453	0.00	480,453	0.79
18	15	5	484,761	0.00	484,761	0.79
19	15	5	488,748	0.00	488,748	0.80
20	15	5	484,446	0.01	484,405	0.80
21	15	5	489,438	0.35	487,722	0.80
22	15	5	486,728	0.01	486,685	0.80
23	15	5	487,370	0.12	486,779	0.80
24	15	5	491,080	0.05	490,812	0.80
25	15	10	980,593	0.11	979,468	2.25
26	15	10	978,288	0.02	978,065	2.24
27	15	10	981,725	0.38	978,027	2.25
28	15	10	974,006	0.12	972,797	2.25
29	15	10	977,260	0.01	977,188	2.26
30	15	10	968,362	0.08	967,617	2.25
31	15	10	979,867	0.08	979,114	2.25
32	15	10	984,598	0.09	983,672	2.24
33	30	5	575,313	-0.13	576,039	6.45
34	30	5	569,643	0.27	568,095	6.46
35	30	5	574,622	0.15	573,739	6.46
36	30	5	568,524	0.40	566,248	6.47
37	30	5	556,257	-0.38	558,353	6.48
38	30	5	565,691	-0.07	566,077	6.47
39	30	5	568,229	0.19	567,131	6.46
40	30	5	574,038	0.05	573,755	6.47
41	30	10	1,158,684	-0.39	1,163,222	7.64
42	30	10	1,157,547	-0.34	1,161,521	7.65
43	30	10	1,154,147	-0.24	1,156,918	7.66
44	30	10	1,143,331	-0.23	1,145,918	7.66
45	30	10	1,122,196	-0.38	1,126,432	7.67
46	30	10	1,143,452	0.18	1,141,344	7.70
47	30	10	1,149,764	0.79	1,140,744	7.67
48	30	10	1,166,223	0.41	1,161,437	7.68
			AVG DEV	0.04		

APPENDIX C

IMPLEMENTATION OF PROBLEM GENERATION AND SOLUTION

Notation used is as follows.

D : total demand per period

ML : demand mix level

DL : duplication level

$MP(p, t)$: share of product *p* in the demand mix of period *t*

V_{it} : total incoming flow to department type *i* in period *t*

C_i : total capacity (available process time) of department type *i*

Problem Generation

30 problems varying by product routes and machine processing times are generated according to the parameters, T, N, P, M, L_p, $R_{p,l}$, and t_i as given in Table 5.1. Problem generation procedure is given in Figure 5.1. Output of this procedure is a *problem* set including processing times of machines of 6 department types and processing routes of 8 product types.

Four experimental instances are generated for each of the 30 problems according to the factor combinations given as follows.

D	200, 400
ML	ML1, ML2 as in Table 5.4

Given the demand per period, demand mix and processing routes, the procedure in Figure C.1 generates experimental instances of the problems by calculating the aggregate flow matrices and capacities of departments.

For each problem, four instances are generated by this procedure. Each instance contains two output sets, *modeloutroute* (input to the mathematical model) and *simoutroute* (input to the simulation model). These sets contain the related parameters and variables as follows.

modeloutroute: P, N, T, t_i, C_i, V_{ijpt} (including entrance and exit) and processing route of each product ($R_{p,l}$)

simoutroute: P, N, T, processing route of each product ($R_{p,l}$), D, t_i

Aggregate Flow Matrix Generation

START

for t = 1 to T

for p=1 to P

for i = 0 to N + 1, j = 0 to N + 1

K = 0

if i-j flow exists in route of product p then K = 1

$$v_{ijpt} = K \times D \times MP(p,t)$$

endfor

endfor

calculate $V_{ijt} = \sum_{p=1}^{P} V_{ijpt}$

for j = 0 to N+1

calculate $V_{jt} = \sum_{i=0}^{N} V_{ijt}$

endfor

endfor

END

Total Department Capacities for Mathematical Model

START

for i = 1 to N

$$C_i = \max_t (t_i V_{it})$$

endfor

END

Figure C.1. Experimental instance generation

Optimization

This procedure is applied to each of the four experimental instances of 30 problems according to 16 (4x4) factor combinations given as follows.

DL	DL1, DL2, DL4 and DL8
r	1, 15, 50 and ∞ for DL8 *
*values are given as relocation cost per machine and they are scaled for each duplication level according to the number of machines per department duplicate	

Given the *modeloutroute*, *initial layout* and *SA settings* sets, the procedure given in Figure C.2 generates the mathematical model and applies DSALAB to solve the instance.

START

for i = 1 to N and n = 1 to numeric part of *DL*
    $C_{ni} = D_i$ / numeric part of *DL*
endfor

generate the mathematical model according to *DL*, relocation cost of a department duplicate (depending on *DL*) and *modeloutroute* set.

define the size of the DDFLP according to *DL*

execute DSALAB with settings for *DL* given in Table 5.3 to improve the initial layout

END

Figure C.2. Optimization Procedure

initial layout : Each duplication level has its own initial layout. The same initial layout is used for all runs of the same duplication level. An *initial layout* set includes: *DL*, X and Y scale of DDFLP and initial X and Y coordinates of department duplicates over periods. Duplicates are numbered consecutively, for example for DL2, duplicates of department type 1 are numbered as 1 and 2, duplicates of department type 2 are numbered as 3 and 4, and so on.

SA settings: There are 16 different settings (4 duplication levels and 4 relocation costs) for each experimental instance. An *SA settings* set contains: α and A values, random number seeds for different sources of randomness, terminating condition (maximum number of moves allowed), unit flow cost per unit distance (1 for all cases) and department duplicate relocation cost.

There are 48 machines on an 8x6 grid layout in all cases (details are given in Figure 5.2 and Table 5.5). Optimization procedure, given in Figure C.2, is applied to 64 experimental settings of 30 problems.

An output set named *simout* is generated to be used in simulation. There are 64 *simout* sets for all factor combinations of each problem. This set includes: *DL*, X and Y scale of layout, location of department duplicates as X and Y coordinates over periods (duplicates are again numbered consecutively, for example for DL2, duplicates of department type 1 are 1 and 2, duplicates of department type 2 are 3 and 4, and so on) and v_{nimjpt} .

Simulation

There are 64 factor combinations for 30 problems (2 total period demand, 2 demand mix, 4 relocation cost and 4 duplication levels). All the instances are simulated for 30 replications using the simulation model described in Chapter 4. We have used two different period length values for D = 400 instances and one period length value for *D* = 200 instances. Given the *simoutroute* and *simout* sets, experiment file of a simulation run is generated. Some key issues are listed as follows.

Number of resources: $DL \times N$

Number of stations: $DL \times N + N$ (one for each department duplicate which is a resource and one for each department type since processing routes are defined using department types).

Number of queues: $DL \times N + T$ (one for each resource and one for each period for the transporter).

Distance matrix for resources: Generated from location of department duplicates.

Processing route sequences: $R_{p,l}$

PL: Period length calculated as 400 x 5 x 55 / 48 = 2291.67 (to be multiplied by 2 or 3)

Arrival rate: EXPO(2*PL* / *D*) for *D* = 200, 400 and EXPO(3*PL* / *D*) for *D* = 400

Part type: Assigned according to a discrete distribution constructed according to ML1 or ML2

Selection of the department duplicate to go in the processing route: Assigned according to a discrete distribution constructed from v_{nimjpt}.

Results for 30 replications are stored in a spreadsheet file and in SIMAN summary report. SIMAN report contains average values of 30 replications in addition to individual replication results. Performance measures in the output files are for all periods and average of periods 2-9. These measures are utilizations for department duplicates and department types, full and empty utilization of the transporter, WIP (total and accumulated by the transporter), flow time (including process time, waiting time in queue and transportation time) of each product type and interperiod transition times.

APPENDIX D

DSALAB RESULTS

Figures D.1 through D.4 summarize averages of cost components over 30 problems in DSALAB solutions. Total, full travel, empty travel and relocation cost values are plotted against duplication levels for four combinations of the demand and demand mix level. Note that, as *r* changes, cost values do not change significantly. This is mainly because the number of relocations decreases as *r* increases and the overall relocation cost remains relatively constant. We observe that, as the duplication level increases empty travel cost slightly increases. However, the reduction in full travel cost component is much more significant, resulting in a reduction in the total cost.

Average number of machine relocations is plotted against duplication levels in Figure D.5 for different period demand, demand mix and relocation cost levels.

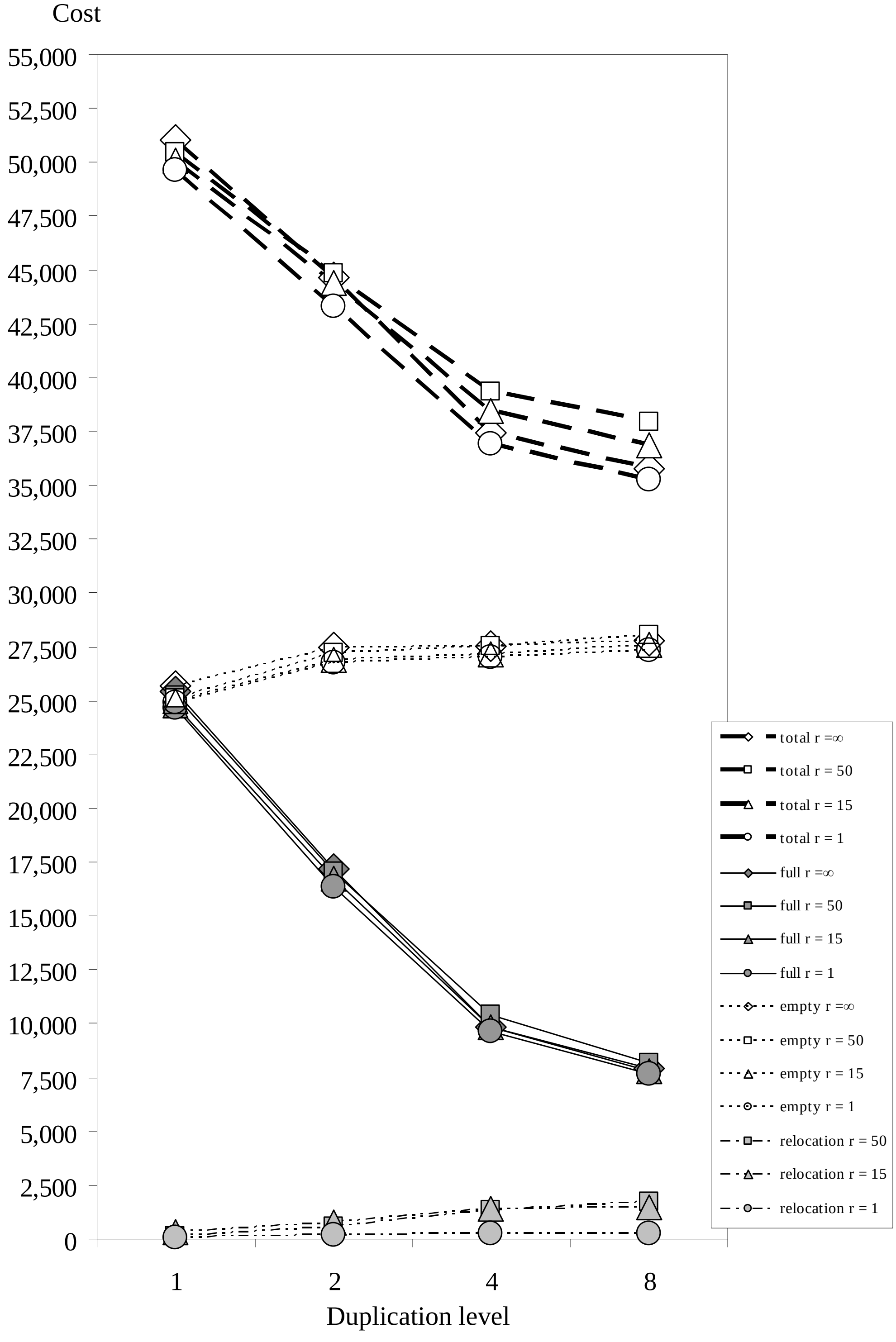


Figure D.1. Cost components vs. duplication level for demand = 200 and mix level ML1

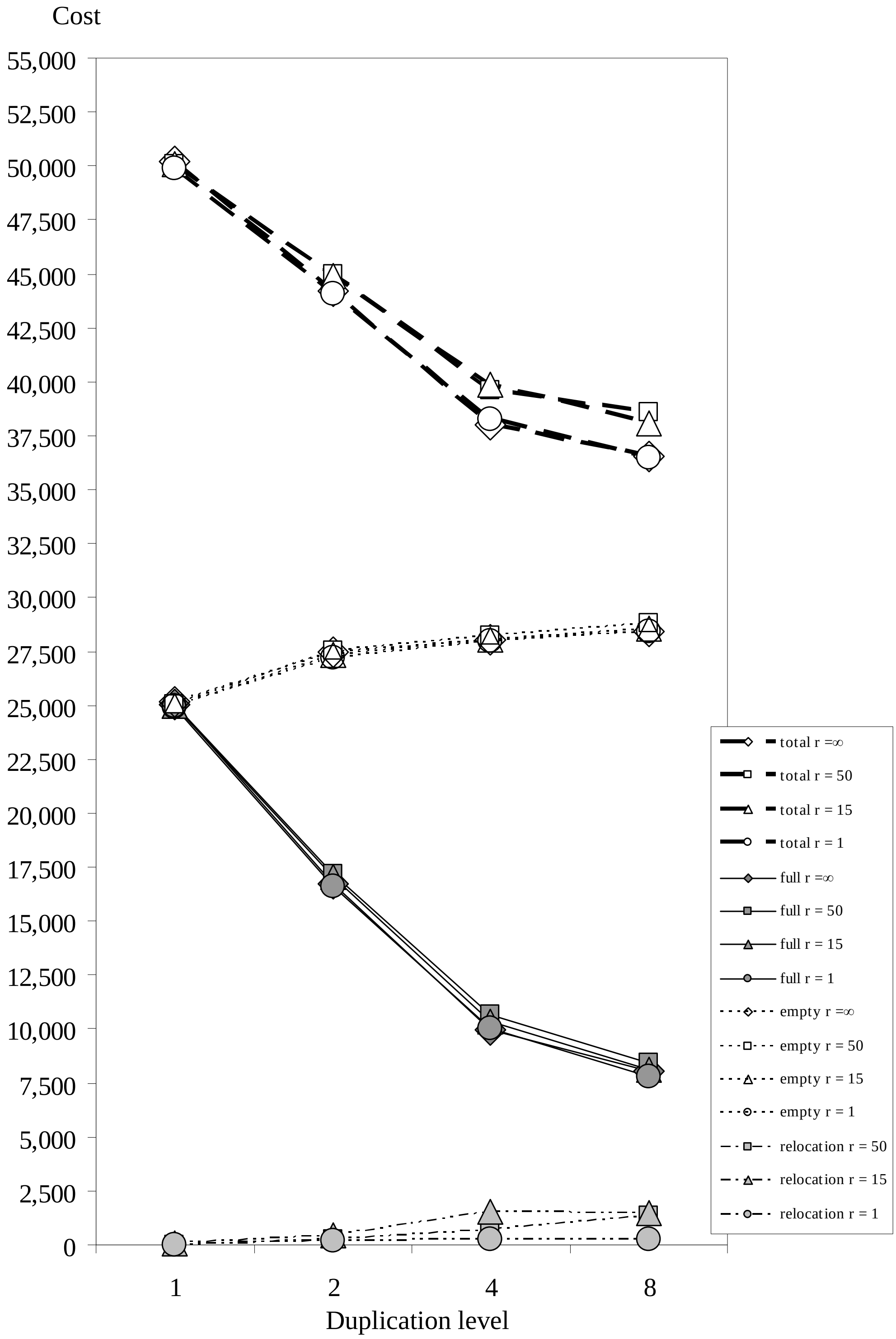


Figure D.2. Cost components vs. duplication level for demand = 200 and mix level ML2

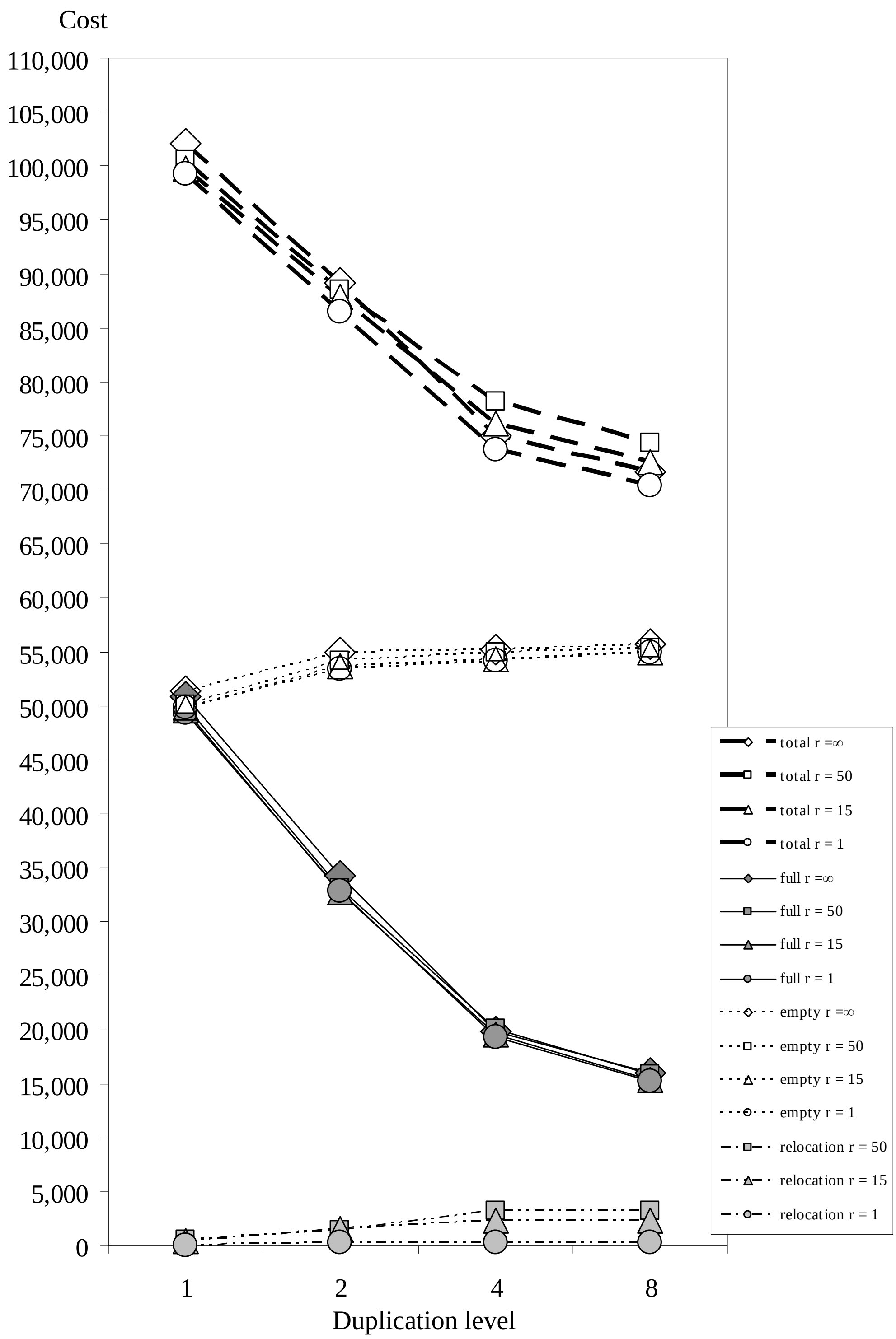


Figure D.3. Cost components vs. duplication level for demand = 400 and mix level ML1

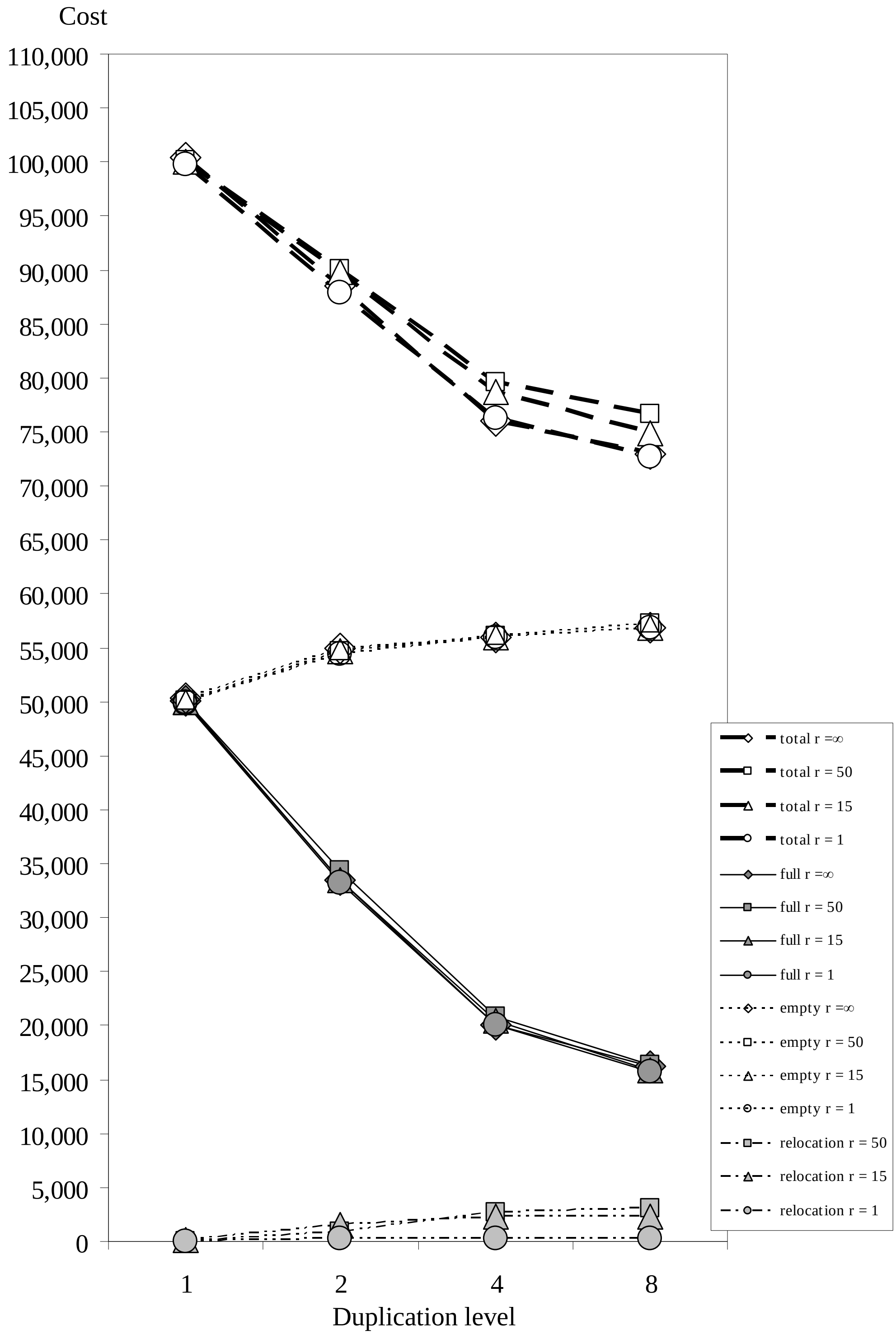


Figure D.4. Cost components vs. duplication level for demand = 400 and mix level ML2

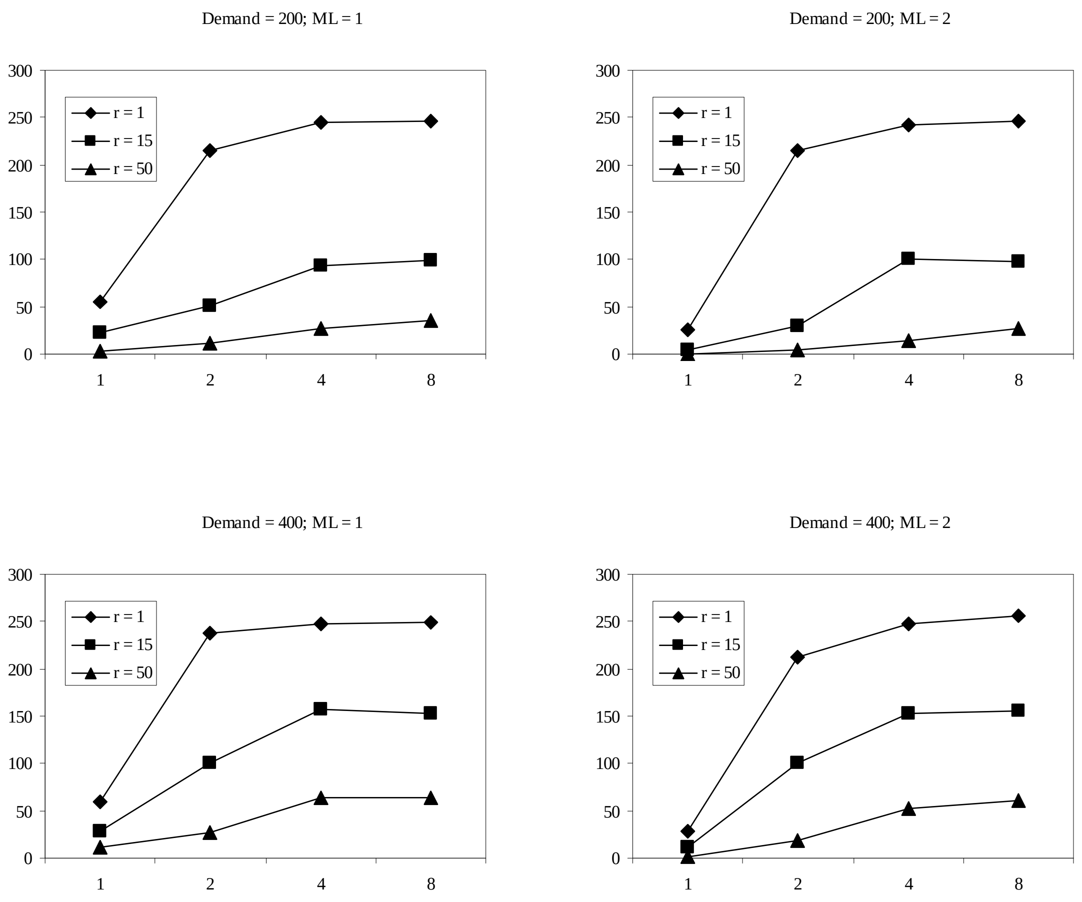


Figure D.5. Number of machine relocations vs. duplication level

APPENDIX E

SIMULATION RESULTS

Simulation results of 30 problems for all factor combinations are reported in Tables E.1 through E.4. Notation used in these tables is as follows.

DEMAND: Demand per period; ML: Demand mix level; DL: Duplication level

IPT %: Ratio of interperiod transition time to period length

MAX DEPT UTIL %: Average of utilizations of the maximally used duplicates of all department types. For each problem instance, the utilization of the maximum used duplicate in a period is found for all department types and average is found over department types given as follows.

$$mdept_i = \max_{n,t}(Util_{n,t,i}) \qquad \text{and} \qquad MAXDEPTUTIL\% = \underset{i}{avg}(mdept_i)$$

$Util_{n,t,i}$ is the utilization of the n^{th} duplicate of department type i in period t. This column contains average and maximum of *MAXDEPTUTIL%* over 30 problems.

TRANSP UTIL %: Transporter utilization for full and empty travel

WIP: Work-in-process inventory

DEVIATION %: Percentage deviation of material handling cost found in the DSALAB solution from the one estimated with simulation (ABS stands for absolute deviation)

PL: Period length used for the arrival rate

TOTAL COST: Total layout cost including full and empty travel cost according to simulation and machine relocation cost

SIM PERIOD: Average realized period length according to the simulation run. It is the time interval between last departure of the previous period’s demand and last departure of the current period’s demand.

Table E.1. Simulation results of 30 problems for r = 1

			IPT %		MAX DEPT UTIL%		TRANSP UTIL % FULL		TRANSP UTIL % EMPTY		WIP		DEVIATION %				
DEMAND	ML	DL	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	ABS	MIN	MAX	TOTAL COST	SIM PERIOD
PL = 4583.33																	
200	1	1	7.05	9.69	37.15	53.43	16.75	24.55	15.90	23.66	13.12	18.03	3.32	1.46	6.34	48,177.49	4,607.50
200	1	2	7.40	10.54	41.58	57.94	11.41	17.00	17.18	25.46	13.82	20.09	3.03	0.39	7.29	42,236.63	4,597.98
200	1	4	8.78	13.83	47.62	65.05	7.19	11.60	17.40	25.75	16.25	25.68	2.48	-1.27	8.63	36,365.16	4,594.30
200	1	8	12.70	22.08	55.75	73.58	6.26	9.93	17.78	26.49	23.52	41.80	2.96	-5.32	7.80	35,599.10	4,598.36
200	2	1	7.03	10.28	33.22	45.94	16.95	24.54	15.96	23.15	13.02	19.06	3.35	1.37	6.83	48,379.24	4,594.94
200	2	2	7.32	10.99	37.29	50.65	11.59	17.11	17.40	25.46	13.58	20.19	3.25	0.90	8.12	42,843.98	4,599.23
200	2	4	8.42	14.86	43.13	58.12	7.40	11.30	17.95	26.54	15.73	27.95	2.74	-1.12	9.13	37,457.50	4,591.33
200	2	8	11.86	22.33	51.22	66.97	6.29	9.53	18.38	26.91	22.07	42.59	2.79	-5.31	9.51	36,471.59	4,591.04
400	1	1	35.41	140.17	65.74	87.74	32.75	47.59	32.52	46.80	139.95	610.30	2.02	-6.08	2.83	100,400.71	4,776.11
400	1	2	48.13	145.85	69.95	91.62	25.55	42.99	34.48	49.37	188.37	616.42	7.92	-21.37	2.53	95,295.52	4,889.86
400	1	4	58.07	160.76	74.46	94.87	20.25	39.59	34.99	49.22	226.24	664.72	13.71	-34.59	1.46	88,905.68	4,933.08
400	1	8	73.06	172.36	78.97	95.21	20.36	38.99	35.45	50.61	281.17	702.94	19.42	-37.23	-0.32	91,528.42	5,014.19
400	2	1	38.61	167.21	59.87	79.77	32.84	49.28	32.36	47.86	161.03	727.84	1.66	-6.04	2.82	100,365.06	4,790.27
400	2	2	45.12	177.57	65.30	86.96	25.46	41.35	35.36	52.11	180.67	740.74	6.14	-20.59	2.52	94,744.56	4,810.89
400	2	4	52.95	180.12	70.48	91.25	19.74	33.16	36.05	53.76	207.64	760.97	11.02	-30.93	2.17	88,381.43	4,865.53
400	2	8	66.81	194.98	76.09	95.42	19.63	35.47	36.77	54.97	257.91	818.37	16.33	-35.51	0.53	90,407.46	4,915.66
PL = 6875																	
400	1	1	5.53	16.59	48.05	67.34	22.45	32.80	21.82	32.00	20.73	69.49	2.12	0.67	4.74	97,394.71	6,874.75
400	1	2	6.70	21.36	52.06	72.15	15.31	23.00	23.32	33.94	25.21	87.97	1.99	-2.56	5.42	85,382.94	6,890.18
400	1	4	9.10	28.41	56.89	77.70	9.77	16.67	23.73	34.98	33.87	114.33	2.61	-9.95	5.99	74,113.84	6,895.74
400	1	8	14.92	42.19	63.47	85.51	8.80	16.71	24.32	36.09	54.62	167.03	4.44	-17.85	4.85	73,410.79	6,900.73
400	2	1	5.67	21.35	42.51	58.36	22.65	32.80	21.83	31.45	21.62	94.21	2.14	0.81	4.89	97,860.86	6,878.93
400	2	2	6.90	33.20	46.58	63.45	15.59	23.43	23.86	34.83	26.91	148.32	2.13	-4.65	5.37	86,820.79	6,868.57
400	2	4	8.75	36.06	51.71	69.05	10.11	17.03	24.50	35.86	33.63	153.10	2.57	-11.93	6.68	76,330.59	6,874.76
400	2	8	14.05	60.32	58.30	75.71	8.86	16.93	25.05	36.54	51.73	216.73	3.59	-18.64	5.91	74,941.70	6,880.30

Table E.2. Simulation results of 30 problems for r = 15

			IPT %		MAX DEPT UTIL%		TRANSP UTIL % FULL		TRANSP UTIL % EMPTY		WIP		DEVIATION %				
DEMAND	ML	DL	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	ABS	MIN	MAX	TOTAL COST	SIM PERIOD
PL = 4583.33																	
200	1	1	7.08	9.81	37.11	53.44	16.84	24.95	15.95	23.90	13.12	18.07	3.33	1.30	6.40	48,616.77	4,602.60
200	1	2	7.39	10.69	41.48	58.19	11.51	16.83	17.19	25.23	13.76	20.15	3.61	0.27	7.49	43,030.38	4,606.61
200	1	4	8.73	14.39	47.69	64.95	7.28	11.59	17.45	26.18	16.31	27.11	2.59	-1.18	8.53	37,769.16	4,595.75
200	1	8	12.67	21.65	55.92	73.76	6.34	9.68	17.90	26.58	23.42	40.30	2.91	-5.53	7.41	37,117.85	4,597.23
200	2	1	7.00	10.25	33.20	45.57	17.02	24.83	15.99	23.39	13.02	18.98	3.28	1.51	6.26	48,582.82	4,596.16
200	2	2	7.27	10.78	37.33	49.97	11.74	17.02	17.48	25.53	13.57	20.51	3.81	1.22	8.54	43,403.24	4,597.42
200	2	4	8.45	14.46	43.16	57.11	7.61	11.63	17.95	26.92	15.73	27.61	2.83	-0.35	9.14	39,022.59	4,591.92
200	2	8	11.85	22.26	51.40	67.72	6.47	10.08	18.44	27.53	22.05	41.53	2.69	-4.83	8.22	38,031.96	4,593.03
400	1	1	35.36	139.28	65.71	87.34	32.83	47.37	32.52	46.80	139.62	606.83	2.06	-6.08	3.02	100,968.46	4,779.75
400	1	2	48.37	153.63	69.90	89.63	24.37	40.09	34.43	48.60	189.71	619.87	6.38	-17.98	2.71	94,722.13	4,898.72
400	1	4	57.90	158.51	74.17	93.11	20.38	38.92	35.16	49.99	225.98	655.28	13.68	-32.49	1.63	91,208.38	4,923.54
400	1	8	74.22	180.43	79.10	95.51	20.36	37.90	35.50	49.57	283.25	701.71	19.30	-36.47	-0.03	93,505.70	5,011.97
400	2	1	38.78	167.21	59.84	79.57	32.88	48.97	32.37	47.86	161.70	727.84	1.66	-5.86	2.96	100,629.67	4,792.58
400	2	2	44.54	170.66	65.08	85.61	25.22	40.38	35.47	52.56	179.62	742.57	5.95	-21.22	3.02	96,130.64	4,828.71
400	2	4	53.07	180.42	70.60	91.25	19.91	34.09	36.07	54.35	206.73	769.89	10.84	-30.20	2.40	90,648.26	4,864.33
400	2	8	66.96	191.67	76.14	95.11	19.68	34.93	36.65	53.68	257.44	802.50	16.39	-37.02	0.50	92,664.81	4,932.17
PL = 6875																	
400	1	1	5.56	15.97	48.02	67.39	22.50	32.64	21.83	32.01	20.77	65.67	2.15	0.67	4.74	97,948.39	6,879.72
400	1	2	6.93	27.92	52.14	72.49	15.20	22.91	23.50	34.20	26.21	121.82	2.05	-1.34	6.03	86,634.39	6,880.25
400	1	4	8.97	26.24	57.01	77.20	9.80	16.63	23.83	35.30	33.41	103.87	2.56	-8.24	6.23	76,498.94	6,890.39
400	1	8	14.73	40.57	63.47	83.80	8.74	16.12	24.31	35.87	53.46	160.77	4.17	-16.24	5.21	75,275.89	6,899.80
400	2	1	5.68	21.35	42.52	58.59	22.69	33.06	21.85	31.80	21.59	94.21	2.12	0.87	4.89	98,126.94	6,879.21
400	2	2	6.96	26.91	46.51	63.09	15.64	24.13	23.96	34.99	26.71	117.67	2.20	-3.32	5.59	88,477.58	6,871.46
400	2	4	8.91	36.32	51.75	69.31	10.30	17.58	24.52	36.18	34.25	159.98	2.55	-11.52	6.37	78,774.02	6,869.77
400	2	8	13.61	51.90	58.51	75.63	8.92	17.25	25.04	36.30	51.25	205.75	3.65	-18.22	5.74	77,084.99	6,881.24

Table E.3. Simulation results of 30 problems for r = 50

			IPT %		MAX DEPT UTIL%		TRANSP UTIL % FULL		TRANSP UTIL % EMPTY		WIP		DEVIATION %				
DEMAND	ML	DL	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	ABS	MIN	MAX	TOTAL COST	SIM PERIOD
PL = 4583.33																	
200	1	1	7.09	9.81	37.08	53.44	17.18	24.95	16.06	23.90	13.14	18.07	3.25	1.30	6.40	48,995.62	4,599.47
200	1	2	7.39	10.62	41.65	58.40	11.65	18.98	17.36	25.23	13.79	20.60	3.96	1.57	7.51	43,296.99	4,606.56
200	1	4	8.74	14.44	47.61	65.39	7.51	11.78	17.68	26.27	16.28	27.56	3.33	-0.62	9.71	38,349.84	4,595.78
200	1	8	12.73	21.88	55.88	74.72	6.55	10.03	18.17	27.02	23.50	41.37	3.13	-5.47	8.26	38,096.75	4,594.82
200	2	1	7.01	10.25	33.17	45.57	17.07	24.83	16.02	23.39	13.03	18.98	3.26	1.26	6.26	48,611.91	4,596.06
200	2	2	7.27	11.00	37.36	50.89	11.72	17.80	17.60	26.15	13.56	20.56	4.04	1.80	8.48	43,351.34	4,599.33
200	2	4	8.35	14.38	43.08	57.61	7.56	11.47	18.04	26.46	15.65	27.67	3.84	0.16	11.03	38,342.32	4,600.63
200	2	8	11.92	23.34	51.52	68.74	6.63	9.79	18.62	27.27	22.28	44.67	2.90	-5.18	9.30	38,365.42	4,586.88
400	1	1	35.35	139.05	65.73	87.14	33.09	48.20	32.70	47.27	139.62	603.76	2.00	-6.08	2.92	101,705.04	4,777.70
400	1	2	49.43	149.20	69.40	89.57	22.54	35.92	34.50	49.35	194.80	623.25	3.73	-14.84	2.61	91,830.06	4,911.97
400	1	4	57.61	164.26	74.32	93.08	20.08	38.04	35.30	51.69	223.27	662.95	12.68	-31.94	1.56	92,213.50	4,944.11
400	1	8	73.92	179.56	78.89	94.84	20.25	38.99	35.47	49.96	282.86	736.90	18.45	-36.74	0.12	94,524.22	5,030.34
400	2	1	38.68	165.87	59.85	79.76	32.97	49.50	32.47	48.00	161.42	716.48	1.58	-5.86	2.86	100,696.66	4,786.60
400	2	2	46.01	171.19	64.80	85.27	23.37	40.49	35.14	51.70	182.90	742.35	2.96	-11.07	3.86	92,133.58	4,847.90
400	2	4	53.91	184.66	70.57	92.22	19.61	33.58	35.89	54.35	210.47	746.46	10.08	-27.92	2.59	90,627.67	4,893.14
400	2	8	66.15	192.45	76.16	94.99	19.70	34.14	36.90	53.82	253.40	810.05	15.31	-35.45	0.79	93,196.35	4,903.94
PL = 6875																	
400	1	1	5.57	16.58	48.05	67.28	22.70	33.36	21.95	32.40	20.88	68.94	2.11	0.52	4.74	98,710.66	6,874.23
400	1	2	6.54	20.51	51.95	72.03	15.10	23.07	23.70	34.16	24.52	83.85	2.35	0.45	5.37	86,811.55	6,887.34
400	1	4	9.50	28.45	57.00	76.80	10.03	16.48	24.09	35.97	35.05	114.37	2.70	-7.97	6.19	78,421.46	6,889.53
400	1	8	15.05	46.06	63.45	84.04	9.05	17.13	24.45	35.67	55.31	189.71	4.30	-17.28	5.10	77,155.99	6,903.58
400	2	1	5.65	21.16	42.50	58.59	22.76	33.22	21.90	31.74	21.59	93.77	2.09	0.83	4.89	98,307.75	6,880.50
400	2	2	6.79	30.88	46.59	62.70	15.77	24.22	23.95	34.69	26.66	140.72	2.52	-0.49	6.64	88,094.68	6,867.07
400	2	4	9.18	40.37	51.52	69.28	10.31	16.31	24.56	35.98	34.51	149.93	2.57	-9.69	6.75	79,404.13	6,879.92
400	2	8	13.45	48.26	58.46	75.60	9.07	16.63	25.20	36.63	49.97	182.63	3.32	-15.39	5.69	78,434.73	6,877.75

Table E.4. Simulation results of 30 problems for Static Distributed Layout (r = ∞)

			IPT %		MAX DEPT UTIL%		TRANSP UTIL % FULL		TRANSP UTIL % EMPTY		WIP		DEVIATION %				
DEMAND	ML	DL	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	ABS	MIN	MAX	TOTAL COST	SIM PERIOD
PL = 4583.33																	
200	1	1	7.09	9.81	37.11	53.44	17.27	24.95	16.09	23.90	13.16	18.07	4.47	1.91	10.13	49,050.78	4,598.27
200	1	2	9.20	23.55	50.89	75.05	11.66	17.14	17.10	26.53	16.55	50.63	5.74	2.89	12.65	42,422.46	4,609.63
200	1	4	18.28	99.58	63.42	84.36	6.70	10.41	17.02	25.53	32.19	195.69	7.09	0.56	15.84	35,133.83	4,629.08
200	1	8	26.56	90.47	70.10	88.45	5.37	8.86	17.47	25.36	44.52	174.29	5.66	-0.64	12.57	34,034.15	4,652.52
200	2	1	7.01	10.25	33.17	45.57	17.07	24.83	16.02	23.39	13.03	18.98	3.53	1.36	6.92	48,611.91	4,596.06
200	2	2	7.45	11.40	40.65	54.83	11.38	16.34	17.46	25.68	13.86	21.82	4.66	2.07	8.69	42,382.32	4,595.61
200	2	4	10.03	42.96	50.56	70.01	6.79	10.66	17.76	26.45	18.97	97.29	5.87	2.72	12.22	36,049.65	4,591.78
200	2	8	14.59	47.17	59.33	77.69	5.49	8.05	18.21	26.17	26.59	102.51	5.39	2.83	12.45	34,770.87	4,588.29
400	1	1	35.52	141.70	65.69	87.14	33.61	48.20	32.77	47.27	140.49	615.08	2.13	-3.62	6.52	102,091.68	4,777.75
400	1	2	85.14	202.63	74.22	92.96	20.99	31.67	32.36	50.72	335.16	844.90	2.64	-5.74	6.48	89,495.34	5,228.31
400	1	4	128.34	302.87	81.40	94.68	11.43	17.81	30.57	47.72	494.75	1,255.15	2.93	-5.16	7.72	74,267.89	5,513.61
400	1	8	145.25	272.94	85.31	98.03	8.98	14.79	30.34	48.55	554.12	1,132.12	2.22	-4.31	4.92	71,535.90	5,661.75
400	2	1	38.70	165.87	59.86	79.76	33.02	49.50	32.48	48.00	161.51	716.48	1.74	-5.29	3.69	100,711.03	4,785.89
400	2	2	58.68	178.86	67.09	87.54	21.51	32.16	34.50	51.47	234.12	774.55	2.22	-7.33	4.45	88,917.15	4,943.66
400	2	4	78.19	219.27	74.40	92.80	12.52	20.93	33.96	51.05	299.05	902.24	2.25	-5.69	6.63	76,020.08	5,127.59
400	2	8	97.92	243.10	79.35	95.71	10.00	15.91	34.30	51.57	363.50	973.28	2.00	-4.28	4.32	73,085.30	5,154.34
PL = 6875																	
400	1	1	5.57	16.58	48.07	67.28	23.08	33.36	22.01	32.40	20.85	68.94	3.28	1.15	8.08	99,152.00	6,875.97
400	1	2	17.02	82.30	61.83	86.09	15.50	23.09	23.36	36.28	61.47	379.32	3.92	-3.67	10.07	86,736.06	6,975.33
400	1	4	40.05	185.12	72.28	89.09	8.81	13.84	22.93	34.45	144.15	759.58	4.65	-1.60	12.57	72,257.67	7,097.71
400	1	8	51.73	155.81	76.47	92.84	7.03	11.54	23.29	33.35	179.95	636.47	3.58	-3.75	8.65	70,086.28	7,184.07
400	2	1	5.65	21.16	42.51	58.59	22.77	33.22	21.90	31.74	21.59	93.77	2.36	1.09	5.32	98,299.74	6,881.25
400	2	2	9.92	47.07	50.84	69.19	15.24	21.95	23.88	34.99	39.07	191.48	3.08	-0.03	6.25	86,124.23	6,881.48
400	2	4	17.36	130.47	60.54	81.95	9.07	14.14	24.12	35.76	62.14	530.27	3.80	-0.24	9.35	73,589.56	6,920.51
400	2	8	24.96	134.31	67.52	87.21	7.31	10.75	24.66	35.40	84.52	531.49	3.43	-0.27	9.08	70,852.28	6,919.67

In Figures E.1 and E.2, total transporter utilization and total cost are plotted against duplication levels for all combinations of four relocation cost and two demand mix levels. For the same factor combinations, Figures E.3 through E.5 shows how the WIP changes depending on the duplication levels, the total cost and the total transporter utilization. Note that the transporter utilization represents the travel cost.

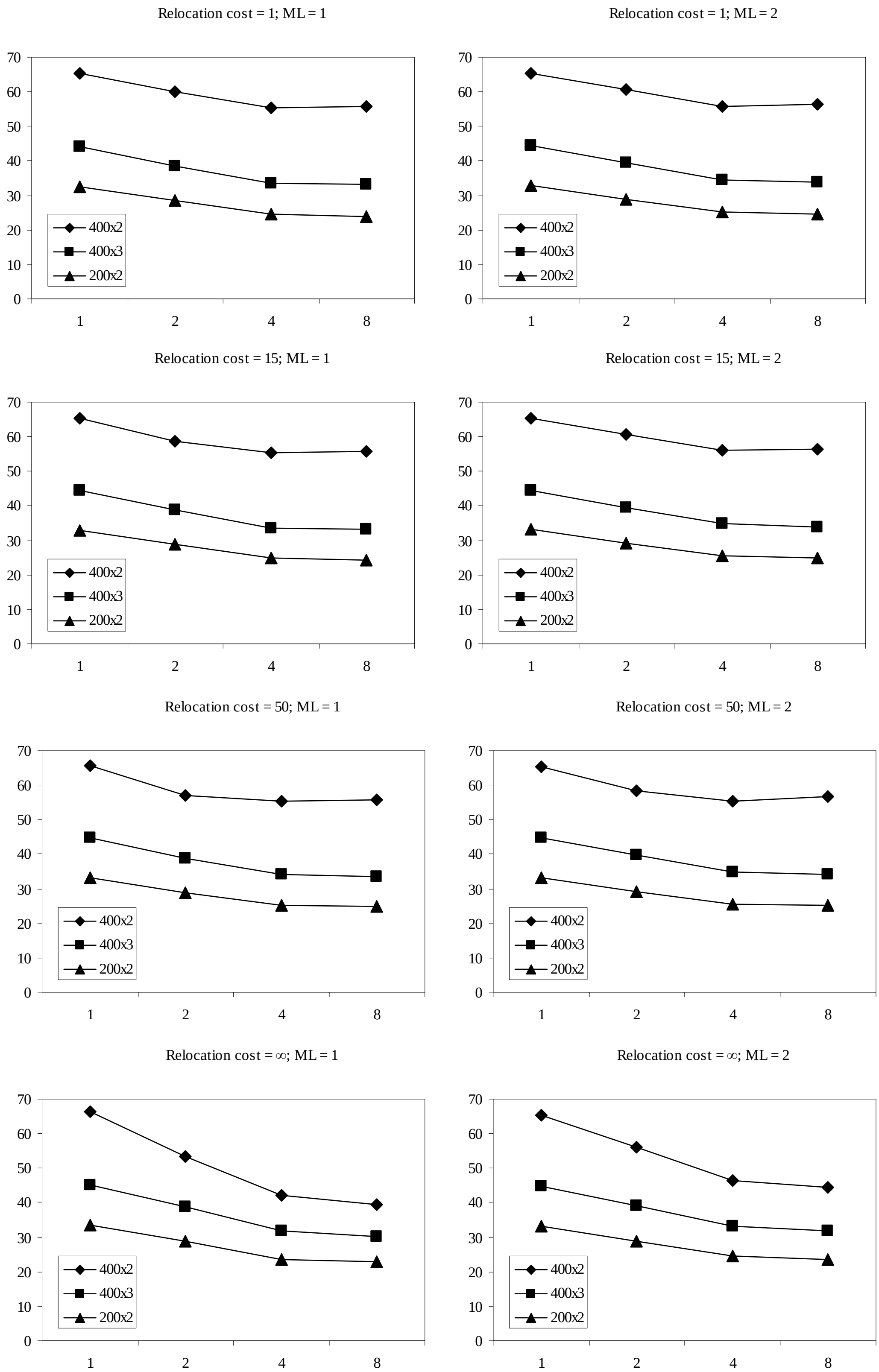


Figure E.1. Total transporter utilization vs. duplication level

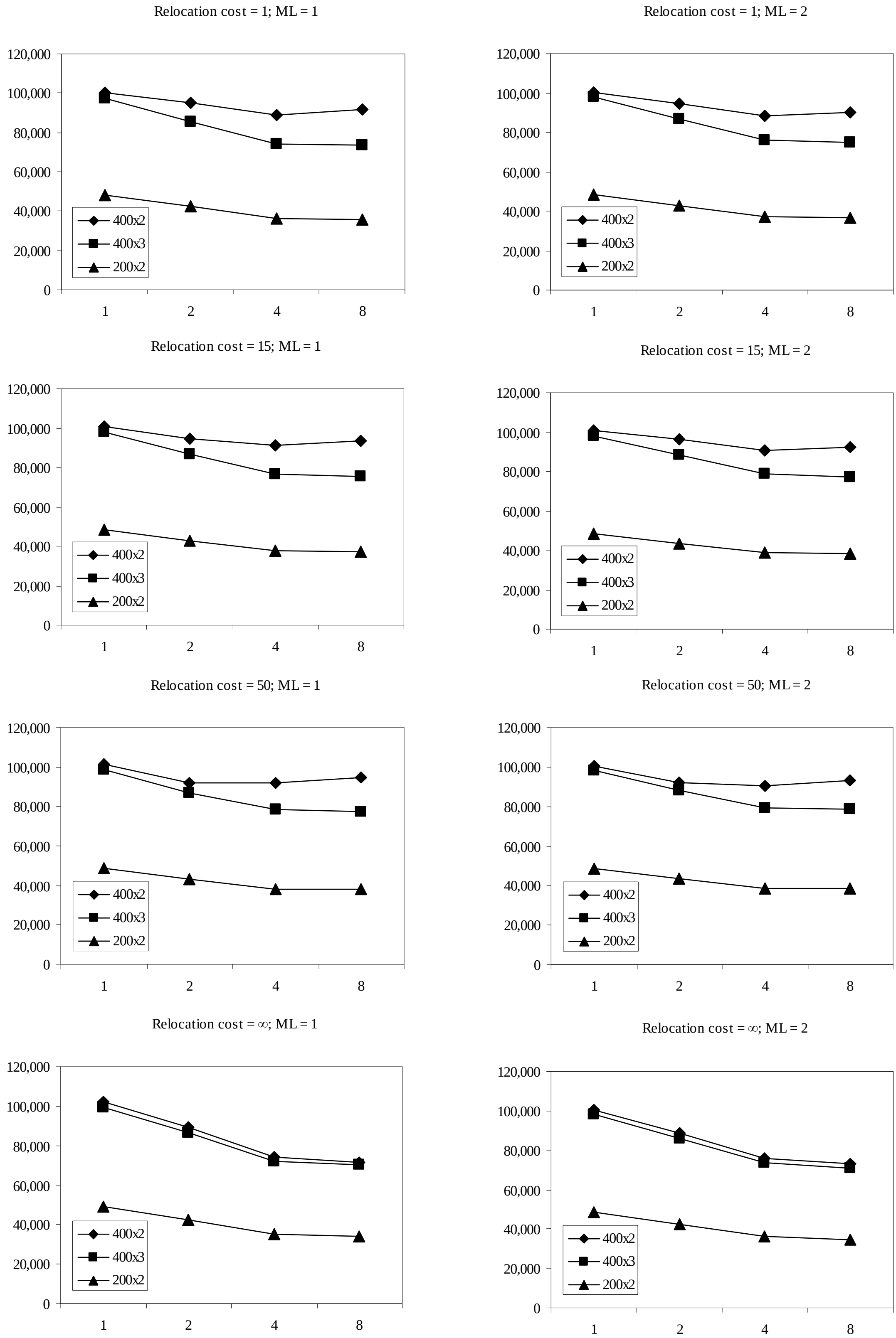


Figure E.2. Total cost vs. duplication level

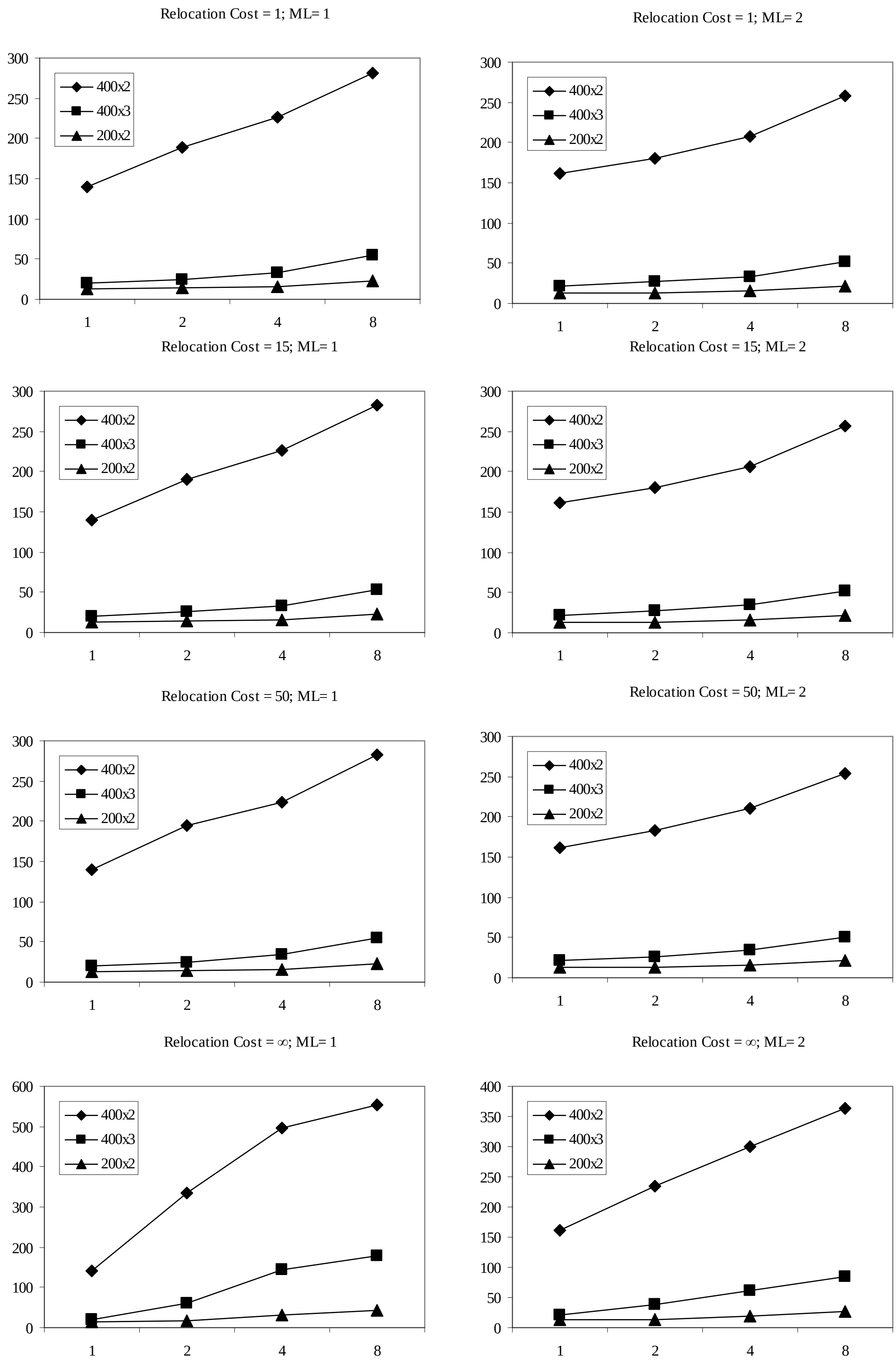


Figure E.3. WIP vs. duplication level

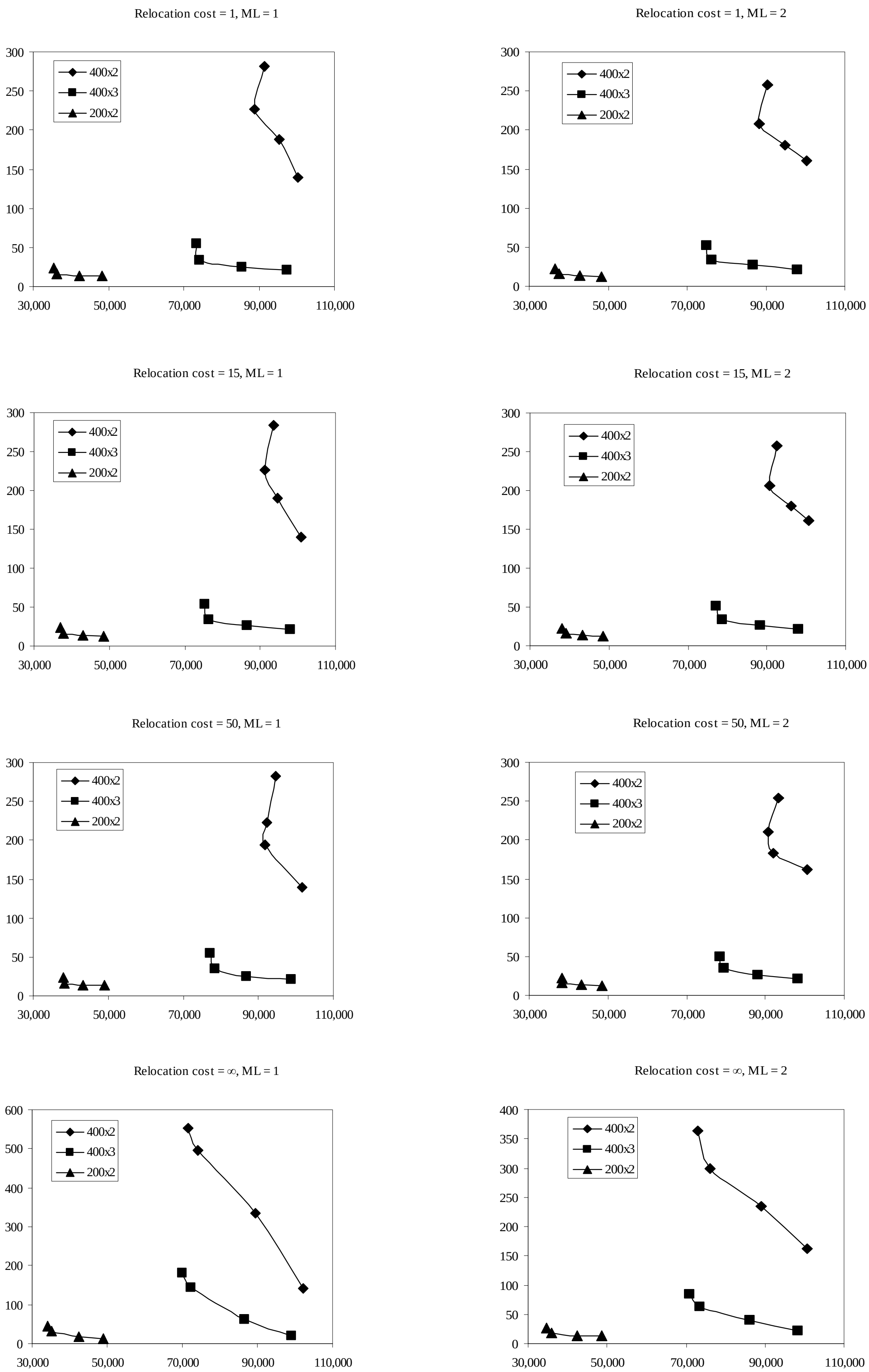


Figure E.4. WIP (vertical axis) vs. total cost (horizontal axis)

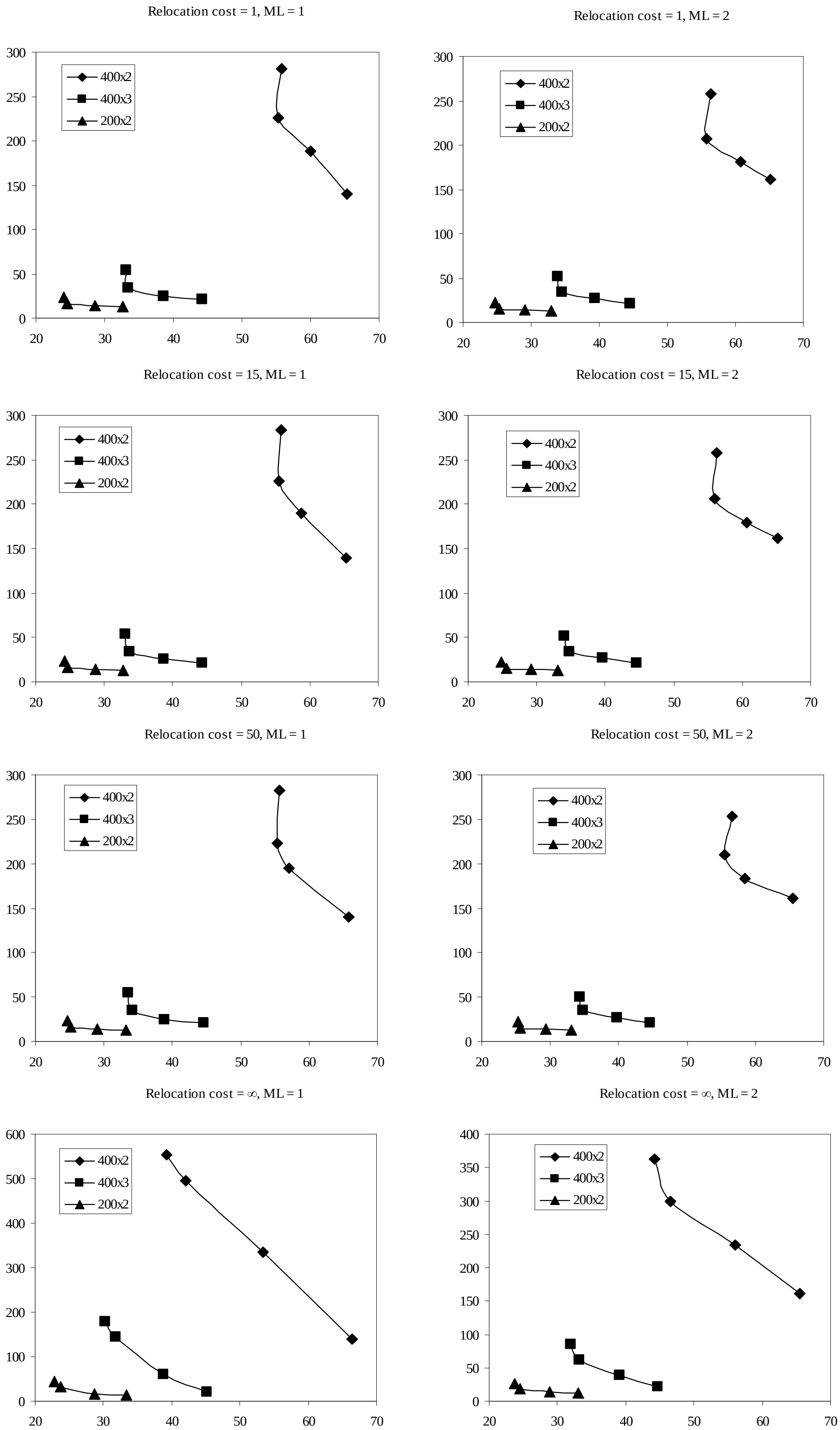


Figure E.5. WIP (vertical axis) vs. total transporter utilization (horizontal axis)