MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN
SULEYMAN DEMIREL UNIVERSITY

UDK 004.021

ZHOLAMANULY OLZHAS

Development of a self-adaptive algorithm and software for routing optimizations

Specialty: "6M070400 – Computing systems and software "

Academic Degree: Master of Computer Sciences

"Admitted to defense":
Dean of Faculty
Assist. Prof., PhD Zhaparov M.
«09» 06 2018г.

Head of Department — Assist. Prof.,PhD. Alimanova M.U.

Scientific advisor — Dr.Tech.Sc., Prof. Amirgaliyev Y.N.

Kaskelen, 2018

ABSTRACT

Nowadays because of globalization and complex supply chain, logistics plays a more important role in developing a successful optimization algorithm. Total logistics costs have become one of the most important economic indicators of logistics efficiency. The total costs included: transportation, order processing or customer service, warehousing, administration and inventory holding costs.

The purpose of this research is optimization of transport logistics using algorithms to optimize finding the optimal path and implementing the basic necessary for the logistics company, taking into account the characteristics of the goods. The goal of the work is to create the optimal algorithm for logistic transportation with customized superstructures for goods delivery.

In this dissertation we will introduce famous logistics optimization algorithms. And try to develop new algorithm which solve main problems of supply chain and try to proof that this new algorithm will be best solution of logistics optimization.

Keywords: logistics, optimizations, algorithm, cost optimizations

АҢДАТПА

Казіргі уакытта жаһандану процессіне байланысты және әлемдік компаниялардың логистиканы толыктай колдануы, тауарлар мен кызметтерді тез және тиімді жолмен жеткізудің кажеттілігі туындап тұр. Тиімді жолмен және аз шығынмен тауарды немесе кызметті жеткізу кез келген компанияның басты максатына айнып отыр. Логистиканын басты мәселелері ол тез, тиімді жолмен, аз шыгынмен жеткізу және қоймаларды дұрыс бақару және тағы баскалары. Осыларды жүйелендіру қажеттілігі дисертацияда қарастырылды.

Бұл жұмыстың негізгі мақсаты оңтайлы жол және тауарлардың сипаттамасын ескере отырып ақпаратты енгізу табу негізінде оңтайландыру алгоритмдерінің көмегімен көлік логистикасын оңтайландыру болып табылады. Яғни, тауарларды жеткізу үшін және логистикалық тасымалдау үшін оңтайлы алгоритмін құру.

Бұл жұмысқа әлемде белгілі қолданыста бар алгоритмдарды талдап, олардың қажетін, әлсіз және мықты жақтарын қарастыру. Және зерттеу нәтижесін ескере отырып жаңа және тиімді алгоритм құру.

Кілт сөздер: Автокөлік, тасымалдау, қиылысу, кептеліс, логистика, механизация, жол торабы, сыйымдылығы, тасжол.

АННОТАЦИЯ

В настоящее время благодаря глобализации и сложной цепочке поставок логистика играет более важную роль в разработке успешного алгоритма оптимизации. Общие логистические издержки стали одним из важнейших экономических показателей эффективности логистики. Общие расходы включали: транспортировку, обработку заказов или обслуживание клиентов, хранение, администрирование и расходы на содержание инвентаря.

Целью исследования данной работы является оптимизация транспортной логистики с помощью алгоритмов оптимизации нахождения оптимального пути и внедрением основных необходимых для логистических компании условии с учетом характеристики груза. Цель работы - создать оптимальный алгоритм для логистических перевозок с настраиваемыми надстройками для доставки грузов или услуг.

В этой работе мы познакомим с известными алгоритмами оптимизации логистики. И попытаемся разработать новый алгоритм, который решает основные проблемы цепочки поставок и попытается доказать, что этот новый алгоритм будет лучшим решением для оптимизации логистики.

Ключевые слова: автомобиль, транспорт, логистика, пробка, пропускная способность, перекресток, автомобилизация, улично-дорожная сеть, магистраль.

CONTENTS

INTRODUCTION

In Kazakhtan industrial and business firms, provision prices are stable throughout the last years, accounting for 11-14% of companies' total prices. the whole transportation prices area unit the largest single provision value part with eight,4% of total value on the average. beside repositing prices, they account for 50-60% of provision value and virtually eight.9% of total value.

On the opposite hand, within the major business areas of the case company, the provision expenses will be as high as twenty five you look after total prices. In these firms the transportation and repositing expenses account for over thirteen of the whole prices. it's necessary to judge logistical processes and their value potency. Optimizing the processes associated with repositing and transporting is that the key to provision value potency. particularly the actions associated with outward-bound provision, that account for a serious a part of transportation prices, have to be compelled to be examined. This analysis takes a better look in one individual case within the case company's provide chain and examines ways that of optimizing total provision value through determination this drawback and optimizing the prices associated with it.

Because of the many existing applications of different vehicle routing problems, a wide variety of researchers and programmers have focused on developing solutions to them. This research described above has shown that existing algorithms based on accurate and heuristic algorithms aims to achieve the lowest transportation cost possible. To our knowledge, no study has considered the time window, step by step planning, using real path instead of straight route and other constraints in the transportation cost model. This features constraints transportation expenses in logistics enterprises, and so it is necessary for logistics to take the this constraints into consideration.

1.1 Supply Chain Management

Supply chain management may be a field of growing interest for each firms and researchers. As nicely told in the recent book by Tayur[1], Ganeshan[2], and Magazine every field has a golden age: This is the time of supply chain management(SCM). The term SCM has been around for more than twenty years and its definition varies from one enterprise to another. We define a provide chain as AN integrated method wherever different business entities like suppliers, makers, distributors, and retailers work along to set up, coordinate, and management the flow of materials, services, and merchandise from suppliers to customers. This chain is consisting with 2 main flows: a forward flow of materials and a backward flow of knowledge. Geunes[3], Pardalos[4], and Romeijn[5] have emended a book that gives a recent review on SCM models and applications.

For many years, researchers and practitioners have centered on the individual processes and entities at intervals the SC. Recently, however, there has been AN increasing effort among the improvement of the whole SC. As companies began realizing the benefits of optimizing the SC joined entity, analysisers began utilizing research techniques to higher model give chains. Typically, a SC model tries to figure out

- the transportation modes to be used,
- the suppliers to be chosen,
- the number of inventory to be management at varied locations among the chain,
- the number of warehouses to be used, and
- the case and capacities of these warehouses.

Following Hax and Candea's treatment of production and inventory systems, the higher than SC selections is classified within the following way:

- Strategic level. These are long-term decisions that have long-lasting effects on the firm such as the number, location and capacities of warehouses and manufacturing facilities, or the flow of material through the SC network. The time horizon for these strategic decisions is often around three to five years.
- Tactical level. These are decisions that are typically updated once every quarter or once every year. Examples include purchasing and production decisions, inventory policies and transportation strategies including the frequency with which customers are visited.

- Operational level. These are day-to-day decisions such as scheduling, routing and loading trucks.

Beamon gave a summary of models in the area of multi-stage supply chain design and analysis. Erengu[6], Simpson[7], and Vakharia[8] surveyed models integrating production and distribution planning in SC. Thomas and Griffin surveyed coordination models on strategic and operational planning.

As a results of the economic process of the economy, however, the models became a lot of advanced. world SC models currently typically attempt to embrace factors like exchange rates, international interest rates, trade barriers, taxes and duties, market costs, and duty drawbacks. All of those factors ar usually difficult to incorporate in mathematical models as a result of uncertainty and nonlinearity. Vidal provided a review of strategic production-distribution models with stress on world SC models. we have a tendency to mentioned a number of the recent models that address the planning and management of worldwide SC networks. Cohen and Huchzermeier[9] additionally gave an intensive review on world SC models. They specialise in the mixing of SC network optimisation with real choices valuation strategies. Most of these SC problems can be modeled as mathematical programs that are typically global optimization problems. Therefore, we next give a brief overview of the area of global optimization.

1.2 Global Optimization

The field of global optimization was initiated during the mid 1980s mainly through the primary works of Hoang Tuy. Since then, and in particular during the last fifteen years, there has been a lot of interest in theoretical and computational investigations of challenging global optimization problems. This has resulted in the development and application of global optimization methods to important problems in science, applied science, and engineering. Exciting and intriguing theoretical findings and algorithmic developments have made global optimization one of the most attractive areas of research. Global optimization deals with the search for a global optimum in problems where many local optima exist. The general global optimization problem is defined by Horst, Pardalos, and Thoai as Definition 1 Given a not empty, closed set $D \subset Rn$ and a continuous function $f : \Omega \rightarrow R$, where $\Omega \subset Rn$ is a suitable set containing D, find least one point $x* \in D$ satisfying $f(x*) \leq f(x)$ for all $x \in D$.

A major difficulty of the global optimization problems is the existence of many local optima. As Horst and Tuy [44] stated, standard local optimization methods are trapped at a local optimum or more generally at a stationary point for which there is not even any guarantee of local optimality. Thus, the use of standard local optimization techniques is normally insufficient for solving global optimization problems. Therefore, more sophisticated methods need to be designed for global optimization problems, resulting in more complex and computationally more expensive methods. Horst and Pardalos gave a detailed and comprehensive survey of global optimization methods. Floudas[10] presented a review of recent theoretical and algorithmic advances in global optimization along with a variety of applications.

In contrast to the objective of the global optimization, the area of local optimization aims at determining a feasible solution that is a local minimum of the objective function f in D (i.e., it is a minimum in its neighborhood, but not necessarily the lowest value of the function f). Therefore, in general, for nonlinear optimization problems where multiple local minima exist, a local minimum (as with any other feasible solution) represents only an upper bound on the one of the global minimum of the main objective function.

In sure categories of nonlinear issues, an area resolution is usually a world one. for instance, during a minimisation downside with a convex (or quasi-convex) objective perform f and a convex possible set D, an area minimizer could be a world resolution (see as an example, Avriel, Horst, Zang and Avriel). it's been shown that many necessary improvement issues will be developed as bursiform minimisation issues. a widely known result by Raghavachari states that the zero-one whole number programming downside is love a bursiform (quadratic) minimisation downside over a linear set of constraints. Giannessi and Niccolucci have shown that a nonlinear, nonconvex whole number program will be equivalently reduced to a true bursiform program underneath the idea that the target perform is finite and satisfies the Lipschitz condition. Similarly, the quadratic assignment downside will be developed as a world improvement downside (see as an example, Bazara and Sherali). In general, additive programming issues square measure love a form of bursiform minimisation downside. The linear complementarity downside will be reduced to a bursiform downside and linear min-max issues with connected variables and linear multi-step bimatrix games square measure reducible to a world improvement downside. The on top of examples indicate yet again the broad vary of issues that may be developed as world improvement issues, and thus make a case for the increasing interest during this space.Global optimization problems remain NP hard for very special cases such as the minimization of a quadratic concave function over the unit hypercube (see for example Garey[11] et al, Hammer[12]), in contrast to the corresponding to main quadratic this problems that can be solved in any time.

Most of the optimization problems that arise in SC are global optimization problems. These problems are of great practical interest, but they are also inherently difficult and cannot be solved by conventional nonlinear optimization methods. Despite the fact that the majority of the challenging and important problems that arise in science, applied science and engineering exhibit non convexities and hence multiple minima, there has been relatively little effort devoted to the area of global optimization as compared to the developments in the area of local optimization. This is partly attributed to the use of local optimization techniques as components of global optimization approaches, and also due to the difficulties that arise in the development of global optimization methods. However, the recent advances in this area and the explosive growth of computing capabilities show great promise towards addressing these issues.

Global optimization methods are divided into two classes, deterministic and stochastic methods. The most important deterministic approaches to nonconvex global optimization are: enumerative techniques, cutting plane methods, branch and bound, solution of approximate sub problems, bilinear programming methods or different combinations of these techniques. Specific solution approaches have been proposed for problems where the objective function has a special structure (e.g., quadratic, separable, factorable, etc.) or the feasible region has a simplified geometry (e.g., unit hypercube, network constraints, etc.).

1.3 Goal and Summary

The focus of this dissertation is to study optimization problems in supply chain operations with cost structures that arise in several real-life applications. A typical feature of the logistics problems encountered in practice is that their cost structure is not linear, due to the presence of fixed charges, discount structures, and different modes of transportation. These price structures haven't been given sufficient attention within the literature, maybe thanks to the difficulty of the underlying mathematical optimisation issues. The goal of this thesis is to develop new optimisation models and algorithms for determination large-scale provision issues with nonlinear price structure.

The supply chain improvement issues we tend to think about square measure developed as giant scale mixed whole number programming issues. The network structure inherent in such issues is employed to develop efficient algorithms to resolve these issues. Due to the scale and difficulty of these problems, the focus is to develop efficient heuristic methods. The objective is to develop approaches that produce optimal or near optimal solutions to logistics problems with fixed charge and piecewise-linear concave cost structures. In this dissertation we also address the generation of experimental data for these optimization models since the performance of heuristic procedures are typically measured by the computation time required and the quality of the solution obtained. Conclusions about these two performance measures are drawn by testing the heuristic approaches on a collection of problems. The validity of the derived conclusions strongly depends on the characteristics of the problems chosen. Therefore, we generated several sets of problems with different characteristics. The outline of the dissertation is as follows. We first introduce the single-item economic lot sizing problem and then discuss extensions to the basic problem to arrive at our problem, which we call the production-inventory-distribution (PID) problem. We present local search based heuristic approaches. We give a brief history of local search and present two approaches for constructing solutions to initiate our local search procedure.

We discuss the complexness of the matter and provides different formulations for issues with fixed charge and piecewise linear cupulate price structures. A dynamic slope scaling procedure (DSSP) is bestowed in section three.3 and a greedy randomised adaptive search procedure (GRASP) is developed in section three.4. DSSP was first introduced by Kim and Pardalos. we have a tendency to refined the heuristic to boost the standard of the solutions obtained. The final section in Chapter three discusses boundary procedures that square measure wont to check the standard of the solutions obtained from native search. The results of intensive machine results square measure bestowed in Chapter four. Details on the planning, implementation, and usage of the subroutines developed are enclosed in Chapter four. Finally, in Chapter five we have a tendency to finish the treatise with a outline of the findings and future analysis directions.

2.1 Single Item Economic Lot Sizing Problem

Many issues in SC optimisation like internal control, production designing, capability designing, etc. ar associated with the straightforward economic ton filler (ELS) downside. Harris is sometimes cited because the first to check ELS models. He thought-about a model that assumes settled demands that occur ceaselessly over time. In 1958, a different approach was projected severally by Manne and by Wagner and Whitin. They divided time into distinct periods and assumed that the demand over a finite horizon is understood before. within the past four decades ELS has received tidy attention and lots of papers have directly or indirectly mentioned this model. Aggarwal and Park gave a quick review of the ELS model and its extensions.

To describe the essential single-item ELS model we are going to use the subsequent notation. Demand (d) for the merchandise happens throughout every of T consecutive time periods. The demand throughout amount t are often satisfied either through production therein amount or from inventory that's carried forward in time. The model includes production of the perform and inventory prices, and to the target is to schedule production to satisfy and be demand at minimum. price|the value|the price} of manufacturing p units throughout amount t is given by r(p) and also the cost of storing It units of inventory from amount t to amount t+1 is h(I). while not loss of generality, we have a tendency to assume each the initial inventory and also the final inventory area unit zero. The mathematical illustration of the ELS model will currently incline as

$$\text{minimize } f(x) = \sum_{t=1}^{T} (r_t(p_t) + \mathrm{h}(I_t)) \quad (1)$$

subject to

$$r_t + I_{t-1} = I_t + d_t \qquad t = 1, \ldots, T, \qquad (2)$$

In the higher than formulation, the first set of constraints needs that the total of the inventory at the beginning of a amount and therefore the production throughout that amount equals the total of the demand throughout that amount and therefore the inventory at the top of the amount. Constraint merely assures that the initial and final inventories ar zero, whereas the last set of constraints limits production and inventory to plus values.

The basic ELS drawback conjointly should be developed as a network flow issues (NFP). This formulation was first introduced by author. The network in Figure 2–1 consists of one supply node and T sink nodes. every sink node needs AN inflow of dt (t = one, 2, ..., T) ANd node D is capable of generating an outflow of dt. for every arc from node D to node t there's AN associated price perform rt(·) for t = one, 2, ..., T, ANd for every arc from node t to node t + one there's an associated price perform ht(·) for t = one, 2, ..., T − 1.

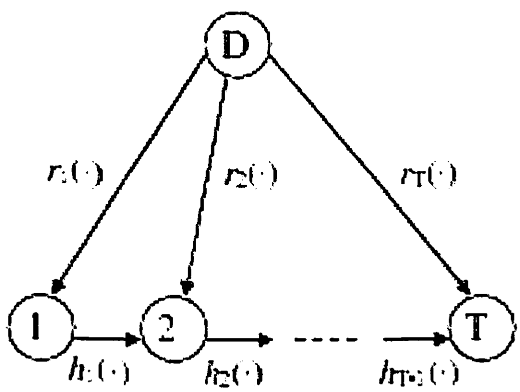

Figure 2–1: The single-item ELS model.

If the price functions rt(·) and ht(·) square measure allowed to be whimsical functions, then the essential ELS downside is sort of difficult to solve; as Florian, Lenstra and Rinnooy Kan have shown it's NP-hard. because of this difficulty and to represent value functions found in follow, sure assumptions square measure typically created concerning the price functions. Aggarwal and Park gave a review of a number of these assumptions and provided improved algorithms for the ELS downside.

We take into account extensions to the essential ELS downside and embody distribution choices within the model. we have a tendency to additionally take into account multiple production plants (facilities) and multiple demand points (retailers). The goal is to fulfill the best-known demand at the retailers through production at the facilities, specified the system wide total production, inventory, and cost is reduced. As mentioned earlier in Chapter one, we have a tendency to tend to hunt recommendation from this downside as a result of the disease downside..

2.2 Production Inventory Distribution Problem

The girdle sickness disadvantage ar developed as a network flow disadvantage on a directed, single offer graph consisting of the various layers. Figure a pair of–2 provides academic degree example with 2 facilities, three retailers and a pair of time periods. each layer of the graph represents a measure. In each layer, a bipartite graph represents the transportation network between the facilities and thus the retailers. Facilities in ordered time periods unit connected through inventory arcs. there is a dummy provide node with offer up to the complete demand. Production arcs connect the dummy provide to each facility in once quantity. usually|this can be} often an easy downside if all costs unit linear. However, many production and distribution activities exhibit economies of scale at intervals that the price of the activity decreases as a result of the amount of the activity can increase. as AN example, production costs typically exhibit economies of scale attributable to fixed production setup costs and learning effects that modify heaps of efficient production as a result of the amount can increase. Transportation costs exhibit economies of scale attributable to the fixed worth of initiating a consignment and thus the lower per unit. shipping price because the volume delivered per cargo will increase.Therefore, we assume

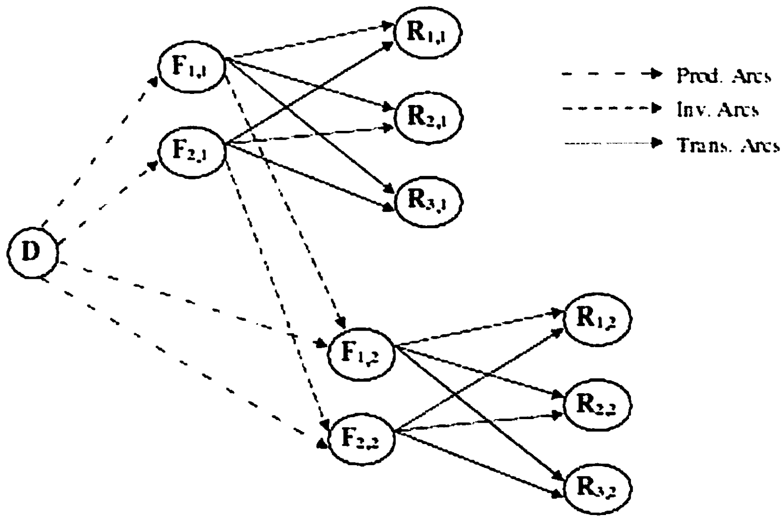

Figure 2–2: A supply chain network with 2 facilities, 3 retailers, and 2 periods.

The production prices at the facilities area unit either of the fixed charge kind or piecewise linear umbilicate kind, the price of transporting product from facilities to retailers area unit of the fixed charge kind, and therefore the inventory prices area unit linear. we tend to additionally create the subsequent simplifying assumptions to the model:

- Backorders don't seem to be allowed.
- Transportation isn't allowed between facilities.
- Products square measure keep at their production location until being transported to a distributer.
- There are not any capability constraints on the assembly, inventory, or distribution arcs.

The first three assumptions can merely be relaxed by adding more arcs among the network and thus the last assumption is justified since the subsequent result by Wagner shows that a network flow drawback with capability constraints is remodeled into a network flow drawback while not capability constraints.

Proposition one each capacitated minimum price network flow drawback on a network with m nodes Associate in Nursingd n arcs is remodeled into the same uncapacitated MCCNFP on an distended network with (n + m) nodes and (n + n) arcs.

Romero Morales thought of similar issues. They assumed that production and inventory prices area unit linear which there's a fixed price of distribution a facility to a distributer. In different words, they accounted for the presence of alleged single-sourcing constraints wherever every distributer ought to be provided from one facility solely. Pardalos and Romeijn[11] and Wu[12] and thought of the multi artefact case wherever there area unit multiple merchandise flowing on the network. assumed production and inventory prices area unit linear and transportation prices area unit of fixed charge kind, whereas Wu dialect and assumed production prices area unit fixed charge and inventory and transportation prices area unit linear.

2.3 Complexity of the PID Problem

The inflammatory disease disadvantage with dish-shaped costs falls below the category of minimum dish-shaped value network flow problems (MCCNFP). Guisewite and Pardalos gave an in depth survey on MCCNFP throughout the primary Nineteen Nineties. it's well-known that even sure special cases of MCCNFP, just like the fixed charge network flow disadvantage or the one offer uncapacitated minimum dish-shaped value network flow disadvantage ar NP-hard (Guisewite and Pardalos). MCCNFP is NP-hard even once the arc costs ar constant, the underlying network is bipartite, or the relation of the fixed charge to the linear charge for all arcs is constant. This has driven the thought of additional structures that might build the matter further tractable.In fact, polynomial time algorithms area unit developed for type of speciallystructured variants of MCCNFP (Pardalos and Vavasis).

The variety of provide nodes and conjointly the quantity of arcs with nonlinear costs affect the difficulty of MCCNFP. it's so convenient to speak to MCCNFP with a fixed selection, h, of sources and fixed selection, k, of arcs with nonlinear costs as MCCNFP(h,k). Guisewite and Pardalos were the first to prove the polynomial solvability of MCCNFP(1,1). Later, powerfully polynomial time algorithms were developed for MCCNFP(1,1) by Klinz and Tuy and Tuy, Dan and Ghannadan. throughout a series of papers by Tuy and Tuy et al. polynomial time algorithms were given for MCCNFP(h,k) where h and k ar constants. it completely was put together shown that MCCNFP(h,k) area unit usually resolved in powerfully polynomial time if min=1.

2.4 Problem Formulation

The problem that we have a tendency to think about could be a multi-facility production, inventory, and distribution drawback. one item is created in multiple facilities over multiple periods to satisfy the demand at the retailers. the target is to attenuate the system wide total production, inventory, and transportation value. Let J, K, and T denote the amount of facilities, the amount of outlets, and therefore the coming up with horizon, severally. Then the ensuing model is minimize

$$\sum_{j=1}^{J}\sum_{t=1}^{T} r_{jt}(p_{jt}) + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T} f_{jkt}(x_{jkt}) + \sum_{j=1}^{J}\sum_{t=1}^{T} h_{jt} I_{jt}$$

Subject to

$$p_{jt} + I_{j,t-1} = I_{jt} + \sum_{k=1}^{K} x_{jkt} \quad j = 1,\ldots,J; t = 1,\ldots,T. \tag{2.4}$$

$$\sum_{j=1}^{J} x_{jkt} = d_{kt} \quad k = 1,\ldots,K; t = 1,\ldots,T. \tag{2.5}$$

$$p_{jt} \leq W_{jt} \quad j = 1,\ldots,J; t = 1,\ldots,T. \tag{2.6}$$

$$I_{jt} \leq V_{jt} \quad j = 1,\ldots,J; t = 1,\ldots,T. \tag{2.7}$$

$$x_{jkt} \leq U_{jkt} \quad j = 1,\ldots,J; k = 1,\ldots,K; t = 1,\ldots,T. \tag{2.8}$$

$$p_{jt}, I_{jt}, x_{jkt} \geq 0 \quad j = 1,\ldots,J; k = 1,\ldots,K; t = 1,\ldots,T. \tag{2.9}$$

Constraints (2.4) and (2.5) square measure the flow conservation constraints and (2.6), (2.7), and (2.8) square measure the capability constraints. If Ujkt, Vjt, and Wjt square measure massive enough then the matter is effectively uncapacitated. As we tend to discovered earlier MCCNFP has the combinatorial property that if it's associate optimum resolution, then there exists associate optimum resolution that's a vertex of the corresponding possible domain. A possible flow is associate extreme flow (vertex) if it's not the bulging combination of the other possible flows. Extreme flows are characterised for potential exploitation in finding the MCCNFP. A flow is extremal for a network flow drawback if it contains no positive cycles. A positive cycle in associate uncapacitated network could be a cycle that has positive flow on all of its arcs. On the opposite hand, for a drag with capability constraints on arc flows, a positive cycle could be a cycle wherever all of the arcs within the cycle have positive flows that square measure strictly but the capability. this suggests that for the PID drawback with unlimited capability associate extreme flow could be a tree. In alternative words, associate optimum resolution exists during which the demand of every merchant are satisfied through just one of the facilities.

2.4.1 Fixed Charge Network Flow Problems

In the fixed charge case the production cost function r(p) is of the following form:

$$r_{jt}(p_{jt}) = \begin{cases} 0 & \text{if } p_{jt} = 0, \\ s_{jt} + c_{jt}p_{jt} & \text{if } 0 < p_{jt} \leq W_{jt}. \end{cases}$$

where s and c are, respectively, the setup and the variable costs of production.

If the distribution cost function, is also of a fixed charge form then it is given by

$$f_{jkt}(x_{jkt}) = \begin{cases} 0 & \text{if } x_{jkt} = 0, \\ s_{jkt} + c_{jkt}.x_{jkt} & \text{if } 0 < x_{jkt} \leq U_{jkt}. \end{cases}$$

where s and c represent, respectively, the setup and the variable costs of distribution.

Due to the separation of the worth functions at the origin, downside is transformed into a zero – one mixed variety applied mathematics (MILP) drawback by introducing a binary variable for each arc with a fixed charge. forward s > zero, the worth operate r(p), is replaced with

$$r_{jt}(p_{jt}) = c_{jt}p_{jt} + s_{jt}y_{jt}.$$

Similarly, the distribution cost functions can be replaced with

$$f_{jkt}(x_{jkt}) = c_{jkt}x_{jkt} + s_{jkt}y_{jkt}$$

where

$$y_{jt} = \begin{cases} 0 & \text{if } p_{jt} = 0, \\ 1 & \text{if } p_{jt} > 0, \end{cases} \quad \text{and} \quad y_{jkt} = \begin{cases} 0 & \text{if } x_{jkt} = 0, \\ 1 & \text{if } x_{jkt} > 0. \end{cases}$$

The MILP formulation of the problem is as follows:

$$\sum_{j=1}^{J}\sum_{t=1}^{T}(c_{jt}p_{jt} + s_{jt}y_{jt} + h_{jt}I_{jt}) + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}(c_{jkt}x_{jkt} + s_{jkt}y_{jkt})$$

The MILP formulation is employed to find associate best resolution. This formulation is beneficial in follow since the relaxed issues area unit linear value network flow issues. we tend to use the final purpose convergent thinker CPLEX, that employs branch and certain algorithms, to resolve the MILP formulation. this is often done to determine the difficulty of finding associate best resolution.

2.4.2 Piecewise-linear Concave Network Flow Problems

If the cost functions in problem (P1) are piecewise-linear concave functions (Figure 2–3), then they have the following form:

$$r_{jt}(p_{jt}) = \begin{cases} 0 & \text{if } p_{jt} = 0, \\ s_{jt,1} + c_{jt,1}p_{jt} & \text{if } 0 < p_{jt} \leq \beta_{jt,1}, \\ s_{jt,2} + c_{jt,2}p_{jt} & \text{if } \beta_{jt,1} < p_{jt} \leq \beta_{jt,2}, \\ \vdots & \\ s_{jt,l_{jt}} + c_{jt,l_{jt}}p_{jt} & \text{if } \beta_{jt,l_{jt}-1} < p_{jt} \leq \beta_{jtl_{jt}}, \end{cases}$$

where β for i = 1, 2, ..., l − 1 are the break points in the interval (0, W), β, l =W, and l is the number of linear segments of production cost function t(·). Due to the concavity of the cost functions the following properties hold:

- $c_{jt,1} > c_{jt,2} > \ldots > c_{jt,l_{jt}}$
- $s_{jt,1} < s_{jt,2} < \ldots < s_{jt,l_{jt}}$

We also assume c, 1 > 0 and s, 1 > 0, since these are production costs.

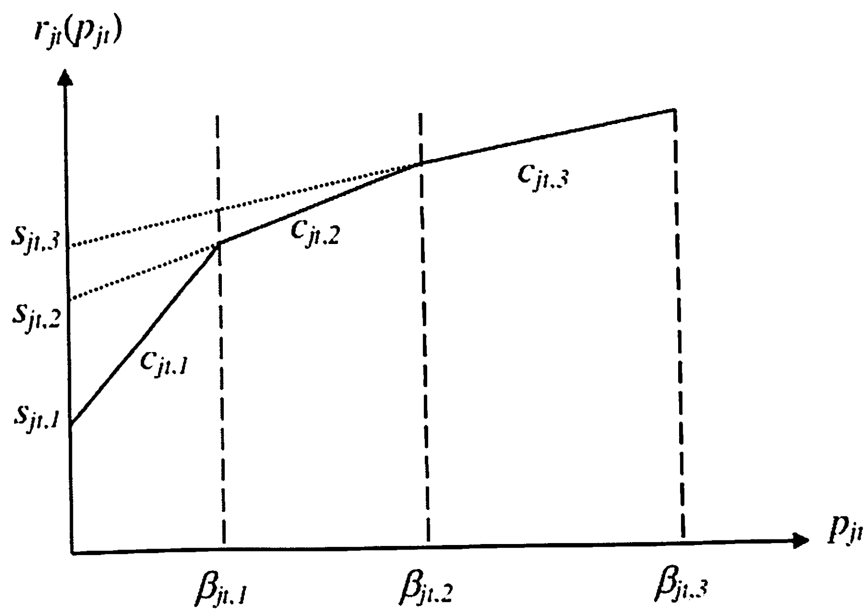

Figure 2–3: Piecewise-linear concave production costs.

Similarly, if the transportation cost functions are piecewise-linear and concave, they have the following form:

$$f_{jkt}(x_{jkt}) = \begin{cases} 0 & \text{if } x_{jkt} = 0, \\ s_{jkt,1} + c_{jkt,1}x_{jkt} & \text{if } 0 < x_{jkt} \leq \beta_{jkt,1}, \\ s_{jkt,2} + c_{jkt,2}x_{jkt} & \text{if } \beta_{jkt,1} < x_{jkt} \leq \beta_{jkt,2}, \\ \vdots & \\ s_{jkt,l_{jkt}} + c_{jkt,l_{jkt}}x_{jkt} & \text{if } \beta_{jkt,l_{jkt}-1} < x_{jkt} \leq \beta_{jkt,l_{jkt}}. \end{cases}$$

The PID problem with piecewise-linear concave costs can be formulated as a MILP problem in several different ways. Below we give four different MILP formulations for the problem: λ-formulation, slope-formulation, ASP formulation, and NSP formulation.

λ-formulation. Any p value on the x-axis of Figure 2–3 can be written as a convex combination of two of the adjacent break points, i.e., Note that at most two of the λ, i values can be positive and only if they are adjacent. This formulation can be used to model any piecewise-linear function (note necessarily concave) by introducing the following constraints.

Note also that we have O(l) (not O(2l)) possibilities for each z vector. In other words, only one of the components of the vector z = (z,1, z,2, . . . z,l) will be 1 and the rest will be zero. The production cost functions can now be written in terms of the new variables. An additional binary variable, y, is added to take care of the discontinuity at the origin

$$\lambda_{jt,0} \leq z_{jt,1},$$

$$\lambda_{jt,1} \leq z_{jt,1} + z_{jt,2},$$

$$\lambda_{jt,2} \leq z_{jt,2} + z_{jt,3},$$

$$\vdots$$

$$\lambda_{jt,l_{jt}-1} + z_{jt,l_{jt}-1} \leq z_{jt,l_{jt}},$$

$$\lambda_{jt,l_{jt}} \leq z_{jt,l_{jt}},$$

$$\sum_{i=1}^{l_{jt}} z_{jt,i} = 1, \text{ and}$$

$$z_{jt,i} \in \{0, 1\}.$$

Note also that we have O(l) (not O(2l)) possibilities for each z vector. In other words, only one of the components of the vector z = (zjt,1, zjt,2, . . . z, l) will be 1 and the rest will be zero. The production cost functions can now be written in terms of the new variables. An additional binary variable, y, is added to take care of the discontinuity at the origin

$$r_{jt}(p_{jt}) = s_{jt,1}y_{jt} + \sum_{i=1}^{l_{jt}}(r_{jt}(\beta_{jt,i}) - s_{jt,1})\lambda_{jt,i}$$

The new variable, y, must equal 1 if there is a positive amount of production at plant j during period t and it must be 0 if there is no production. This is handled by the following constraints: y ≥ (1 − λ,0) and y ∈ {0, 1}. The distribution cost functions can similarly be modelled by introducing new y, z, and λ variables. The PID problem with piecewise-linear concave production and distribution costs can now be written as

$$\sum_{j=1}^{J}\sum_{t=1}^{T}(s_{jt,1}y_{jt} + h_{jt}I_{jt} + \sum_{i=1}^{l_{jt}}(r_{jt}(\beta_{jt,i}) - s_{jt,1})\lambda_{jt,i})$$

$$+\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}(s_{jkt,1}y_{jkt} + \sum_{i=1}^{l_{jkt}}(f_{jkt}(\beta_{jkt,i}) - s_{jkt,1})\lambda_{jkt,i})$$

Slope-formulation. The slope-formulation is analogous to the λ-formulation within the sense that the p values on the coordinate axis of Figure 2–3 area unit once more rewritten in terms of the break points. However, this point p isn't a convexo-concave combination of the break points. However, during this case one or a lot of of the elements of z may be one. If one in all the elements is one (e.g. z ,m=1), then all of the preceding elements should equal one (z ,i = one for all i = one, . . . , m − 1). The slope-formulation of the inflammatory disease drawback is given by

$$\sum_{j=1}^{J}\sum_{t=1}^{T}\left(s_{jt,1}y_{jt}+h_{jt}I_{jt}+\sum_{i=1}^{l_{jt}}\Delta r_{jt,i}\gamma_{jt,i}\right)+\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}\left(s_{jkt,1}y_{jkt}+\sum_{i=1}^{l_{jkt}}\Delta f_{jkt,i}\gamma_{jkt,i}\right)$$

ASP formulation. Kim associate degreed Pardalos used an Arc Separation Procedure (ASP) to remodel network flow issues with piecewise-linear indented prices to network flow issues with fixed charges. exploitation identical procedure, the inflammatory disease downside with piecewise-linear indented prices is remodeled into a fixed charge network flow downside. every arc is separated into ljt arcs as shown in Figure 2–4. All of the new arcs have fixed charge value functions.

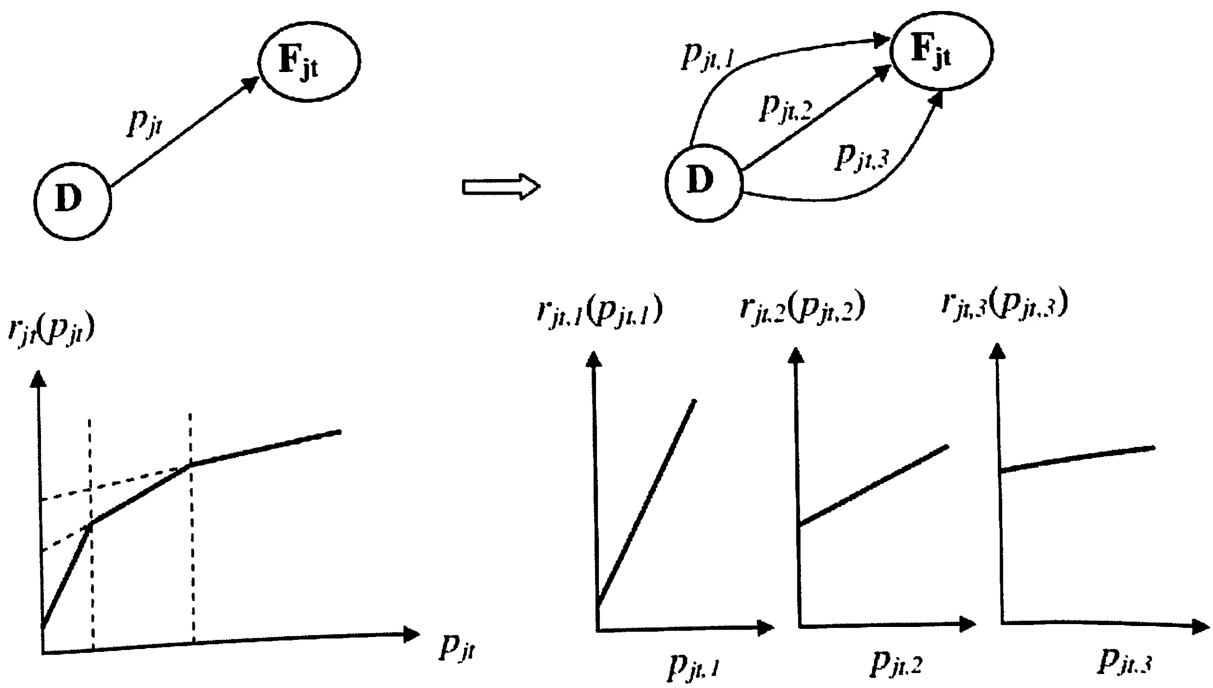

Figure 2–4: Arc separation procedure

The network grows in size after the arc separation procedure. The number of production and transportation arcs in the extended network is given by

$$\sum_{j=1}^{J}\sum_{t=1}^{T} l_{jt} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T} l_{jkt}.$$

The production cost can now be written in terms of the cost functions of the new arcs created in the following way:

$$r_{jt}(p_{jt}) = \sum_{i=1}^{l_{jt}} r_{jt,i}(p_{jt,i}) = \sum_{i=1}^{l_{jt}}(c_{jt,1}p_{jt,1} + s_{jt,1}y_{jt,1}).$$

Note that the equality may not hold in general without additional constraints to restrict the domain for each separated arc cost function.

However, for concave cost functions the equality holds due to the properties given by. Kim and Pardalos gave a simple proof based on contradiction. This formulation isclosely related to the λ-formulation in the sense that only one of the linear pieces will be used in an optimal solution. As the z variables in the λ-formulation, the vector of y variables in the below formulation will have O(l) possible values. The MILP formulation after the arc separation procedure is

$$\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{i=1}^{l_{jt}}(c_{jt,i}p_{jt,i} + s_{jt,i}y_{jt,i}) +$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{i=1}^{l_{jt}}(c_{jkt,i}x_{jkt,i} + s_{jkt,i}y_{jkt,i}) + \sum_{j=1}^{J}\sum_{t=1}^{T} h_{jt}I_{jt}$$

NSP formulation. We developed a Node Separation Procedure (NSP) to rework the PID downside with piecewise-linear Associate in Nursingd depressed prices to PID issues with fixed charge prices in an extended network. The NSP procedure is comparable to the ASP procedure, however it leads to a bigger network as a result of the NSP separates the inventory arcs furthermore. though the network grows in size, this formulation is helpful as a result of its applied mathematics (LP) relaxation ends up in tight lower bounds. Once {the downside|the matter} is remodeled into a fixed charge network flow problem through NSP, the lower bounding procedure explained in section three.5 is wont to get smart lower bounds. Figure 2–5 illustrates the node separation procedure.

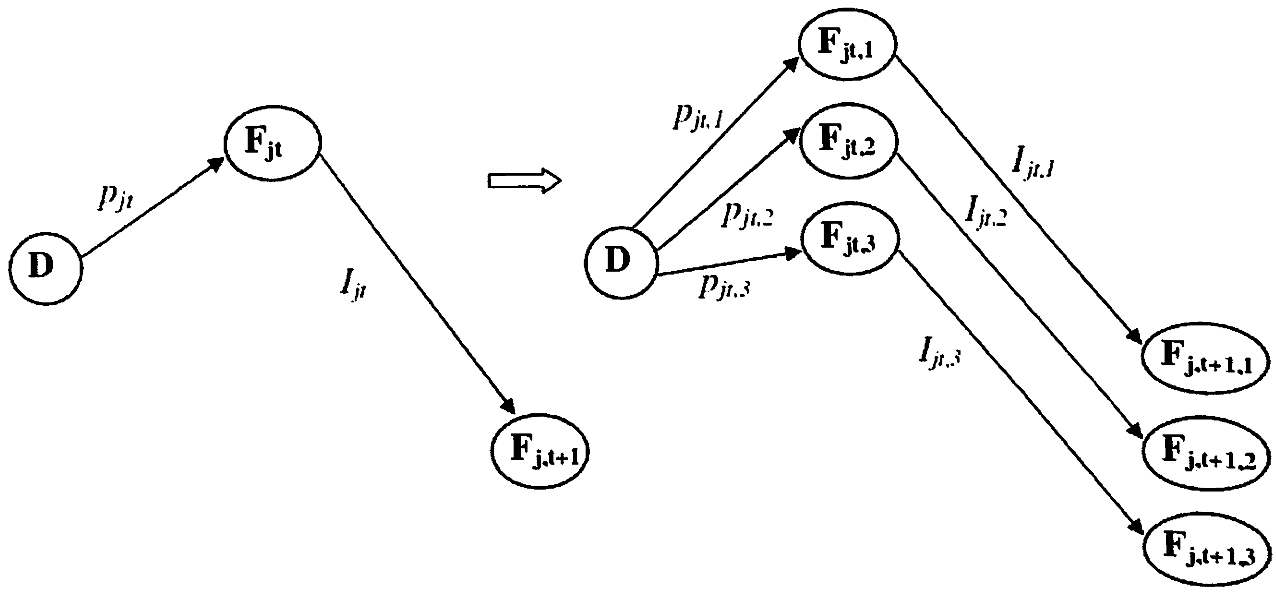

Figure 2–5: Node separation procedure.

The MILP formulation for PID problems with piecewise-linear concave costs,linear inventory costs, and fixed charge distribution costs after the node separation procedure is

$$\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{i=1}^{l_{jt}}(c_{jt,i}p_{jt,i} + s_{jt,i}y_{jt,i} + h_{jt}I_{jt,i}) + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{i=1}^{l_{jkt}}(c_{jkt}x_{jkt,i} + s_{jkt}y_{jkt,i})$$

SOLUTION PROCEDURES

The inflammatory disease downside, as represented in section a pair of.4, is developed as a zero – one Mixed whole number Linear Program (MILP) because of the structure of the assembly and cost functions. In fact, most of the precise answer approaches for combinatorial optimisation issues remodel the matter into the same zero – one MILP and use branch and certain techniques to unravel it optimally. 2 of the newest branch and certain algorithms for fixed charge transportation issues were conferred by McKeown and Ragsdale and Lamar and Wallace. Recently, Bell, Lamar and Wallace conferred a branch and certain formula for the capacitated fixed charge transportation downside. Most of the branch and certain algorithms use conditional penalties known as up and down penalties, that contribute to finding smart lower bounds.

Other actual answer algorithms embody vertex enumeration techniques, dynamic programming approaches, cutting-plane strategies, and branch-and-cut strategies. though actual strategies have matured greatly, the procedure time needed by these algorithms grows exponentially because the downside size will increase. In several instances they're ineffective to provide associate best answer efficiently. Our procedure experiments indicate that even for moderate size inflammatory disease issues the precise approaches fail to find an answer.

The NP-hardness of the matter motivates the utilization of approximate approaches. Since the inflammatory disease downside with fixed charge or piecewise-linear cup-shaped prices falls beneath the class of MCCNFP, it achieves its best answer at associate extreme of the possible region (Horst and Pardalos). Most of the approximate answer approaches exploit this property. Some recent heuristic procedures are developed by Holmqvist, Migdalas and Pardalos , Diaby , Khang and Fujuwara , Larsson, Migdalas and Ronnqvist , Ghannadan et al., and Sun Recently, Kim and Pardalos provided a dynamic slope scaling procedure (DSSP) to unravel FCNFP. refined the DSS and presented results for bipartite and layered fixed charge network flow problems.

A wide kind of the heuristic approaches for NP-hard issues square measure supported native search. native search ways offer a framework for looking the answer area specializing in native neighborhoods. within the following we have a tendency to gift native search primarily based resolution ways for the PID drawback.

3.1 Local Search

Local search relies on an easy and natural technique that's most likely the oldest improvement technique, trial and error. However, native search algorithms have verified to be powerful tools for backbreaking combinatorial improvement problems, specifically those known as NP-hard. The book altered by Aarts and Lenstra may be a very good offer that gives some applications nonetheless as quality results.

The set of solutions of associate improvement disadvantage which is able to be visited by an {area|a district|a region|a locality|a vicinity|a part|a section} search formula is known as the search area. Typically, the potential region of the matter is defined as a result of the search space. However, if generating potential solutions is not straightforward, then a different search space is additionally defined, that takes advantage of the special structures of the matter. If a look space with the exception of the potential region is utilized, then the target operate got to be modified so as that the impracticability of a given answer is identified.

A basic version of native search is repetitive improvement. in several words, a general native search formula starts with some initial answer, S, and keeps replacement it with another answer in its neighborhood, N(S), until some stopping criterion is satisfied. Therefore, to implement a section search rule the following ought to be identified:

- a neighborhood, N(S),
- a stopping criterion,
- a move strategy,
- associate degreed Associate in Nursing analysis perform (this is alone the target perform if the search space is that the doable region).

Good neighborhoods sometimes profit of the special combinatorial structure of downside and unit sometimes drawback dependent. The rule stops If the current answer does not have any neighbors of cheaper price (in the case of minimization). the elemental move strategy in native search is to maneuver to associate improved answer among the neighborhood. usually two strategies unit enforced that unit celebrated as: the first higher move strategy and so the most effective admittible move strategy. among the first higher move strategy, the neighboring Associate in Nursingswers unit investigated associate passing a really pre-specified order and so the first answer that shows an improvement is taken as a result of consecutive answer. The order throughout that the neighbors unit searched might affect the solution quality and so the procedure time. among the most effective admittible move strategy, the neighborhood is searched totally and so the most effective answer is taken as a result of consecutive answer. throughout this case, since the search is thorough the order of the search is not very important.

Lately, completely different heaps of delicate strategies square measure developed to allow the search to escape from domestically optimum solutions among the hopes of finding higher solutions. These delicate procedures unit spoken as metaheuristics among the literature. among the subsequent we tend to tend to first provides a transient history of native search beside an inventory of metaheuristics, then discuss the procedure quality of native search, and finally offer our native search procedure for the pelvic inflammatory disease drawback.

3.1.1 History of Local Search

The use of native search in combinatorial optimisation features a long history. Back in late Nineteen Fifties, lager beer and Croes resolved travelling salesman issues (TSP) mistreatment edge-exchange native search algorithms for the first time. Later, sculptor Refined the edge-exchange algorithms for the TSP and bestowed 3-exchange and Or-exchange neighborhood functions. Reiter and Sherman examined numerous neighborhoods for the TSP and introduced the multi-start strategy. after, Kernighan and sculptor bestowed a variable-depth search rule for uniform graph partitioning.

sculptor and Kernighan [56] conjointly with success applied a variable-depth search rule for the TSP.

In the Nineteen Eighties a lot of generalized approaches were planned that combined native search with different heuristic algorithms. These approaches permit moves to solutions that don't essentially offer higher objective perform values. samples of these subtle approaches, known as metaheuristics, are: simulated hardening, search, genetic algorithms, neural networks, greedy randomised adjustive search procedure (GRASP), variable neighborhood search, hymenopter systems, population heuristics, memetic algorithms, and scatter search. Recent reviews on these procedures is found within the book altered by Pardalos and Resende.

3.1.2 Complexity of Local Search

An important question a few native search formula is that the variety of steps it takes to succeed in a domestically optimum resolution. The time it takes to go looking the neighborhood of a given resolution is sometimes polynomially delimited. However, the amount of moves it takes to succeed in a neighborhood optimum from a given resolution might not be polynomial. as an example, there ar TSP instances associate degreed initial solutions that the native search takes an exponential variety of steps underneath the 2-exchange neighborhood. to investigate the quality of native search Johnson, Papadimitriou and Yannakakis[15] introduced a quality category referred to as PLS (polynomial-time native search). PLS contains issues whose neighborhoods is searched in polynomial time. He provides an intensive survey for the speculation of PLS-completeness.

It is necessary to tell apart between the quality of a neighborhood search drawback and a neighborhood search heuristic. A native an area a neighborhood search drawback is that the drawback of finding native optima by any means that whereas a neighborhood search heuristic is finding local optima by the quality repetitive procedure. a noteworthy and typical example is applied math (LP). For record native optimality coincides with world optimality and also the Simplex technique is viewed as a neighborhood search formula. The move strategy of a neighborhood search ormula affects the period. Similarly, the pivoting rule affects the period of the Simplex echnique. it's well-known that within the worst case the Simplex technique could take ssociate degree exponential variety of steps for many pivoting rules. it's still associate legree open drawback whether or not there exists a pivoting rule that makes the Simplex technique a polynomial time formula. However, record is resolved by alternative ways like the ellipsoid or interior purpose algorithms in polynomial time.

In the past twenty years there has been significant work on the speculation of native search. many native search issues are shown to be PLS-complete, however the quality of finding native optima for several fascinating issues remains open, though machine results are encouraging.

3.1.3 Local Search for the PID Problem

In this section, we tend to explain concerning the neighborhood, the move strategy, the stopping criterion, and also the analysis operate for the inflammatory disease drawback, that area unit the most ingredients of an area search formula. The inflammatory disease drawback is developed as a network flow drawback in section a pair of.4, wherever the target is that the diminution of a umbilicate operate. we tend to cash in of the special structure of network flow issues to define a section and a move strategy. we tend to first provide many definitions of neighborhood for MCCNFP and so adopt one among these definitions and supply a section definition for the inflammatory disease drawback. Next, we tend to discuss the move strategy, the stopping condition, and also the analysis operate. The generation of initial solutions are going to be mentioned later in sections three.3 and 3.4.

Neighborhood definitions for MCCNFP. the standard definition of native optimality defines a section of an answer S within the following means wherever could be a vector norm and $\varepsilon > $ zero. underneath this definition of a section an answer S is domestically optimum if it's uphill to decrease the target operate worth by rerouting atiny low portion ε of flow really, sure as shooting umbilicate functions the ε-neighborhood ends up in all extreme flows to be domestically optimum. Therefore, this normal definition of neighborhood, known as the ε-neighborhood, isn't terribly helpful for our drawback.

Proposition a pair of each extreme flow could be a native optimum for associate degree uncapacitated network flow drawback with one supply and with fixed arc prices underneath Nε. Proof. so as to make associate degree ε-neighbor of a given extreme flow allow us to reroute associate degree ε quantity to 1 of the demand nodes. deduct associate degree ε from all the arcs on the trail from the supply thereto demand node. this may not amendment the present value owing to 2 reasons. First, the present flow on all arcs on the trail from the supply thereto demand node is beyond ε so that they can still have positive flow. Second, the arcs have the subsequent value structure:

$$r_{jt}(p_{jt}) = \begin{cases} s_{jt} & \text{if } p_{jt} > 0. \\ 0 & \text{if } p_{jt} = 0. \end{cases}$$

Now flow that ε quantity from a different path. can produce a minimum of one cycle within the network which implies a minimum of one arc that had zero flow will currently have positive flow. Thus, the overall value won't decrease. Gallo and Sodini have conjointly shown that each extreme flow may be a native optimum below Nε for similar issues with the subsequent value structure:

$$r_{jt}(p_{jt}) = \mu p_{jt}^{\alpha}$$

where $0 < \alpha < 1$ and $\mu > 0$.

There area unit different pouch-shaped price functions, however, that each extreme flow isn't essentially a neighborhood optimum underneath Nε. take into account the subsequent example in Figure 3–1 wherever there area unit 2 demand nodes, 2 transhipment nodes, and one supply node.

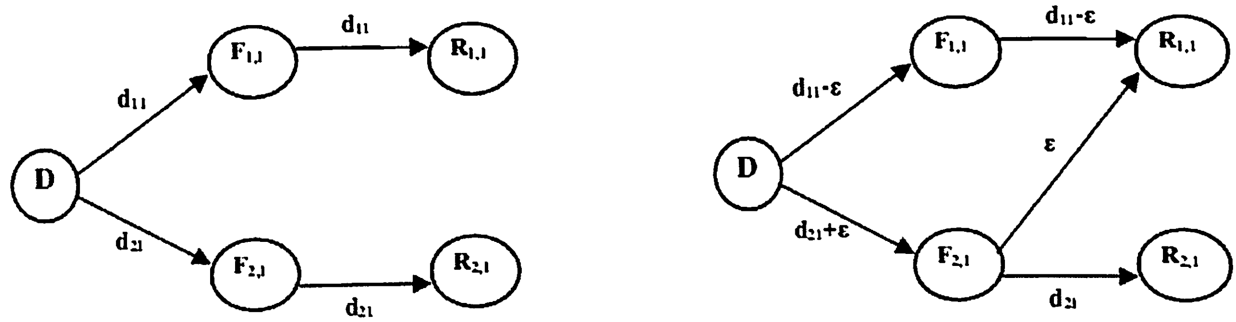

Figure 3–1: An example for ε-neighborhood.

If price functions square measure fixed charge then the intense flow on the left in Figure 3–1 includes a total cost. The non-extreme ε-neighbor on the correct in Figure 3–1 includes a total price. The difference between the prices is ε. If applicable price values square measure chosen then this difference is negative that indicates that the intense flow isn't an area optimum. However, the ε-neighborhood remains not helpful for our downside as a result of it's tasking to look this neighborhood efficiently.

Gallo and Sodini developed the subsequent generalized definition of neighborhood for MCCNFP. Definition three NAEF(S) is possible to the matter associated it's an adjacent extreme flow. Here, 2 extreme flows square measure adjacent if and provided that they differ solely on the trail between 2 vertices. Thus, the graph connection the 2 extreme flows S and Si contains one aimless cycle. Gallo and Sodini delineated a procedure that detects if a given extreme flow may be a native optimum or finds a brand new higher extreme flow by finding a series of shortest path issues. Their procedure constructs a modified network and solves a shortest weighted path downside for every vertex within the current answer, S. The modified network is built to stop moving to nonextreme or distant solutions.

Guisewite[16] and Pardalos[17] developed the subsequent a lot of relaxed definition of neighborhood for MCCNFP. Definition three.3 NAF(S) is possible to the matter associated it's an adjacent flow. 0Under this definition, S is adjacent to the extreme flow S if it is obtained by rerouting a single sub-path within S. NAF(·) is a relaxed neighborhood with respect to NAEF(·) in the sense that adjacent solutions but not only extreme adjacent solutions are included in the neighborhood. In this case it is easier to reach the neighboring solutions because the modified graph that prevented nonextreme solutions is not required and the shortest weighted path problem need not be solved for all vertices of S. It is sufficient to solve the shortest weighted path problem for the branch points (nodes with degree greater than two) and the sink nodes.

Neighborhood definition for the PID problem. A generally accepted rule is that the quality of the solutions are better and the accuracy of the final solutions are greater if a larger neighborhood is used which may require additional computational time. Therefore, it is likely that NAF(·), which is defined above, will lead to better solutions. Thus, we define a neighborhood for the PID problem in the following way which is, in principle, similar to NAF (·).

Definition three answer S could be a neighbor to the present possible solution S if the ability provision in an exceedinglyll amongst in every of} the merchants is modified or if the demand of a retailer comes from constant facility however through production in a different amount. Here, to achieve a neighbor Si from Associate in Nursing extreme answer S we tend to first reckon the demand of a merchant from the present path to it merchant then route it through a different path. Associate in Nursing example is shown in Figure 3–2 wherever the demand of merchant one in amount a pair of (R1,2) at first comes from facility one in amount a pair of (F1,2), however it's rerouted through facility a pair of in amount one (F2,1). Note that dynamical the ability that provides a merchant might not essentially end in Associate in Nursing extreme possible answer. Therefore, when the flow to a merchant is rerouted we want to force the answer to be a tree, which can end in further price savings..

In Figure 3–2, for example, there is a cycle between nodes D, F2,1, and F2,2. This cycle indicates that there is a positive amount of inventory carried forward from period 1 to period 2 at facility 2 and at the same time there is a positive amount of production at facility 2 in period 2. To remove this cycle we should either eliminate the inventory or the production from the cycle. However, we know that rerouting the inventory through node F2,2 will actually increase the total cost and this can easily be shown. When we moved from the extreme point solution on the left to the nonextreme solution on the right rerouting the demand of node R1,2 was cheaper if node F1,2 is used rather than F2,2. Therefore, the production arc should be removed from the cycle. A more formal discussion on cycle detection and elimination is given later in this section.

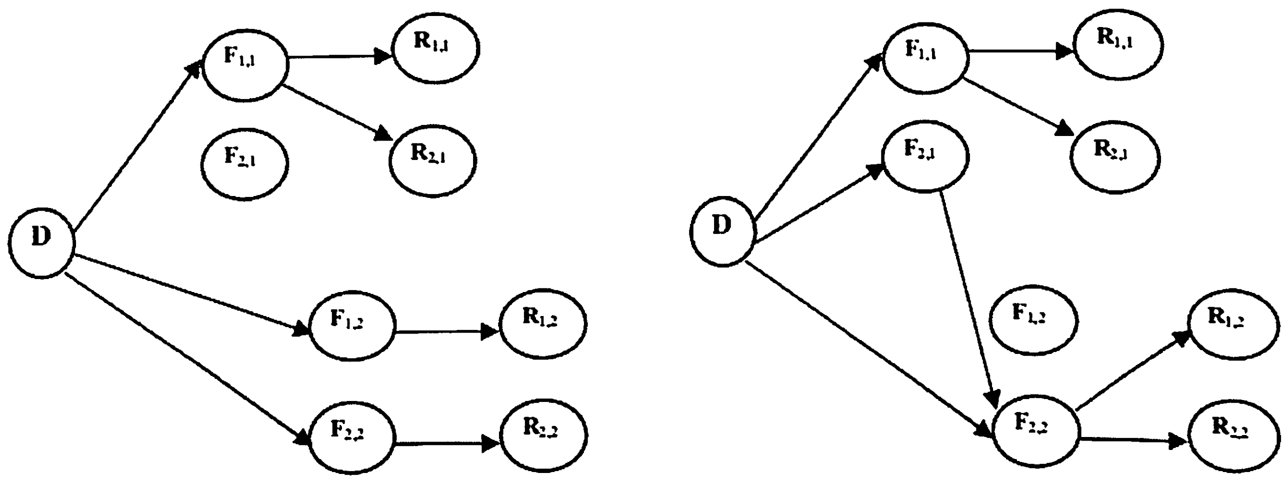

Figure 3–2: An example of moving to a neighboring solution.

The move strategy, the evaluation function, and the stopping Condition. The move strategy determines the order in which the neighboring solutions are visited. As we mentioned earlier, first better and best admissible are the two common move strategies implemented. Guisewite[18] and Pardalos[19] and presented computational results using both move strategies on network flow problems. These authors provided empirical evidence that the first better move strategy requires less time and the quality of the solutions are comparable for both strategies. Therefore, in our local search algorithms we used the first better move strategy. In this strategy, the neighboring solutions are investigated in a pre-specified order and the first solution that shows an improvement is taken as the next solution. The improvements are measured by an evaluation function. The evaluation function that we use calculates the total cost of a solution which is simply the objective function. The local search algorithm terminates when there is no more improvement. In other words, the stopping condition determines whether the current solution is a local optimum or not. If the current solution is a local optimum then the procedure terminates.

The local search algorithm is summarized in Figure 3–3. Let j be the facility that currently supplies retailer k's demand in period t through production in period. Also, let δ be the additional cost of changing the supplier to facility j in period τ ($t \geq \tau$). In other words, δτ is the cost of rerouting the demand of retailer k under consideration. Note that $\delta\tau = 0$ when $j = j$ and $\tau = \tau$. In the local search procedure given in Figure 3–3, j* and τ* are such that

Cycle detection and elimination. Due to the special structure of our network, identifying cycles and removing them can be done efficiently. A cycle for the PID problem means that at least one of the facilities is producing and also carrying inventory in the same period. Therefore, with a single pass over all facilities and time periods the cycles can be identified. The procedure is summarized in Figure 3–4. Note that two types of cycles can be created and they are analyzed separately in the following paragraphs.

Type I cycle. Assume that the demand of retailer k in period t is satisfied through production at facility j at time τ after it is rerouted. If facility j is already carrying inventory from period $\tau - 1$ to period τ, then a type I cycle is created. Only one cycle is created and it can be eliminated by rerouting the inventory. Figure 3–5 gives an example of a type I cycle where the demand of retailer 1 in period 3 is originally satisfied from production at facility 2 in period 3. If it is cheaper to satisfy the demand of retailer 1 in period 3 by production at facility 1 in period 3, then this will result in the network flow given on the right of Figure 3–5. To eliminate the cycle the amount of inventory entering node F1,3 is subtracted from its current path and added on to the production arc entering F1,3.

```
procedure Local
  Let S be an initial extreme feasible solution
  while (S is not a local optimum) do
    for (t = 1,...,T) and (k = 1,...,K) do
      Calculate δ_jτ, j = 1,...,J, τ = 1,...,t
      Δ := min{δ_jτ : j = 1,...,J, τ = 1,...,t}
      if Δ = 0 then S is a local optimum
      else if Δ < 0 then
        Reroute the flow through the new path
        Check for any cycles and eliminate them
      end if
    end for t and k
  end while
  return S
end procedure
```

Figure 3–3: The local search procedure.

```
procedure Cycle
  for (l = t,...,0) do
    if (p_jl > 0 and I_jl > 0) then
      if p_jl is on the new path then reroute I_jl
      otherwise reroute p_jl
  end for
  return S
end procedure
```

Figure 3–4: Cycle detection and elimination.

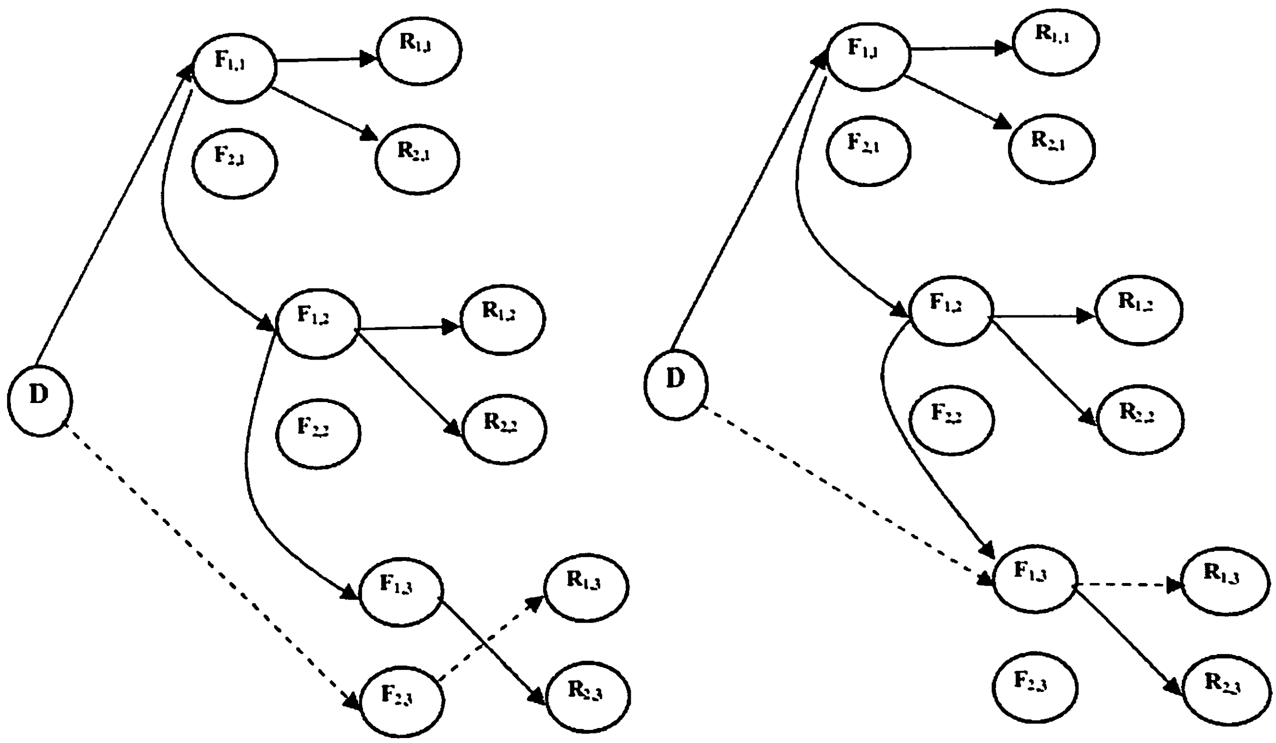

Figure 3–5: An example of a type I cycle.

Type II cycle. Assume that facility j currently produces in several periods and that the demand of retailer k in period t is satisfied through production from an earlier period after it is rerouted. The rerouting of retailer k's demand may result in one or more cycles. Figure 3–6 gives an example of a type II cycle. The flow on the left is the starting extreme flow which indicates that the demand of retailer 1 in period 3 is currently satisfied through production in period 3 at facility 2. Assume that it is actually cheaper to satisfy the demand of retailer 1 in period 3 through production at facility 1 in period 1. This will result in the flow given on the right of Figure 3–6 which has two cycles. To eliminate these cycles the production amounts in periods 2 and 3 at facility 1 are eliminated and carried forward as inventory from period 1.

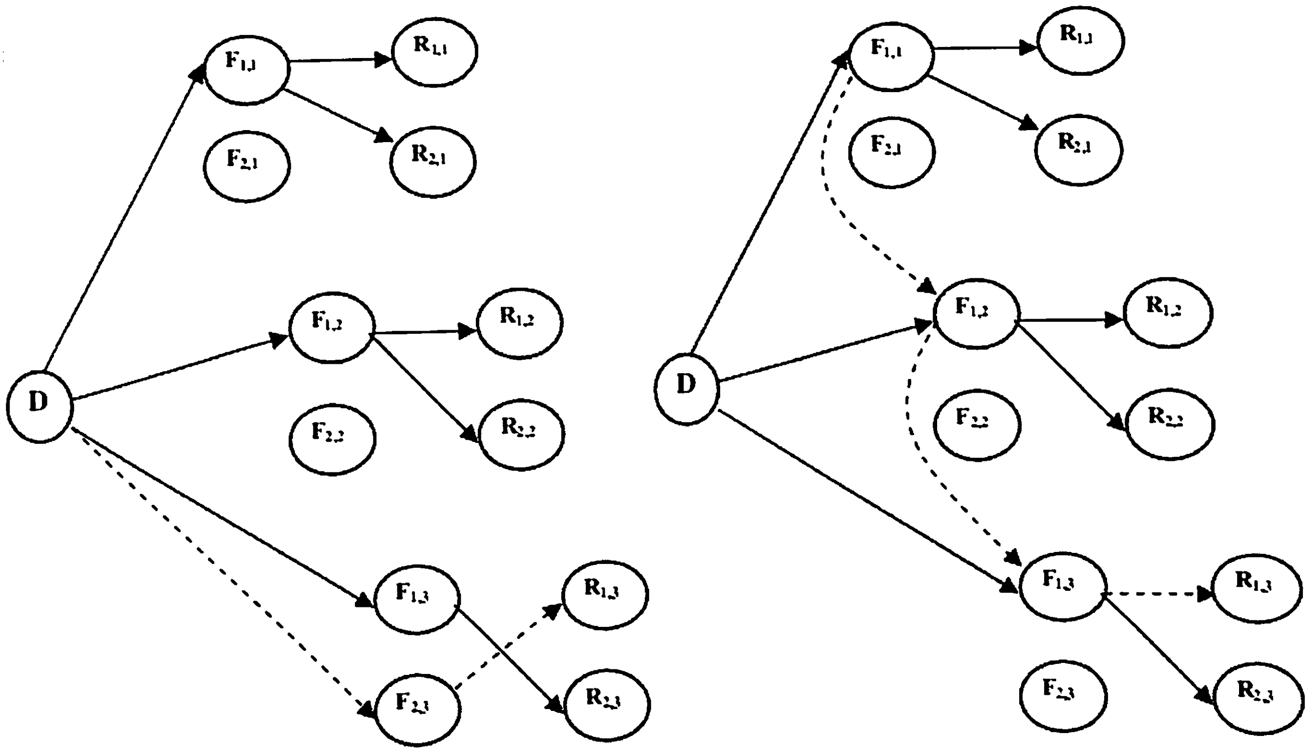

Figure 3–6: An example of a type II cycle.

So far, we have talked about the neighborhood, the move strategy, the evaluation function, and the stopping condition. However, we have not discussed how the initial solutions are found. Generation of initial solutions may be critical for local search procedures. Good starting solutions may lead to better quality local optima. In sections 3.3 and 3.4 we discuss several procedures for generating good initial solutions.

3.2 Dynamic Slope Scaling Procedure (DSSP)

The DSSP is a procedure that iteratively approximates the concave cost function by a linear function, and solves the corresponding network flow problem. Note that each of the approximating network problems has exactly the same set of constraints,

procedure DSSP

$q := 0$ /*set the iteration counter to zero*/

Initialize $\overline{c}_{jt}^{(q)}$ and $\overline{c}_{jkt}^{(q)}$ /*find an initial linear cost*/

while (stopping criterion not satisfied) **do**

$q := q + 1$

Solve the following network flow problem:

minimize $\sum_{j=1}^{J} \sum_{t=1}^{T} (\overline{c}_{jt}^{(q-1)} p_{jt}^{q} + h_{jt} I_{jt}^{q}) + \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{t=1}^{T} \overline{c}_{jkt}^{(q-1)} x_{jkt}^{q}$

subject to original constraints of (P1)

Update $\overline{c}_{jt}^{(q)}$ and $\overline{c}_{jkt}^{(q)}$

if $\text{cost}(S^{(q)}) < \text{cost}(S)$ **then** $S := S^{(q)}$

end while

return S

end procedure

Figure 3–7: The DSSP algorithm.

and differs solely with relevance the target operate coefficients. The motivation behind the DSSP is that the indisputable fact that a cotyloid operate, once reduced over a group of linear constraints, can have Associate in Nursing extreme best answer. Therefore, there exists a linear value operate that yields an equivalent best answer because the cotyloid value operate. Figure 3–7 summarizes the DSSP formula. necessary problems relating to the DSSP formula square measure finding initial linear prices, change the linear prices, and stopping conditions.

Finding Associate in Nursing initial linear value (c(j0)). Kim and Pardalos investigated 2 different ways in which to initiate the formula for fixed charge and piecewise-linear cotyloid network flow issues. Here, we have a tendency to generalize the heuristic for all cotyloid prices with the property that the overall value is zero if the activity level is zero. In different words, rjt(·) and fjkt(·) square measure cotyloid functions on the interval $[0, \infty)$ and rjt(0) = fjkt(0) = zero. The initial linear value factors we have a tendency to use.

Updating scheme for c(jqt). Given a feasible solution S(q) = (p(q), I(q), x(q)) in iteration q, the objective function coefficients for the next iteration are expressed in linear form as

$$\bar{c}_{jt}^{(q)} = \begin{cases} r_{jt}(p_{jt}^{(q)})/p_{jt}^{(q)} & \text{if } p_{jt}^{(q)} > 0, \\ \bar{c}_{jt}^{(q)} & \text{if } p_{jt}^{(q)} = 0. \end{cases}$$

$$\bar{c}_{jkt}^{(q)} = \begin{cases} f_{jkt}(x_{jkt}^{(q)})/x_{jkt}^{(q)} & \text{if } x_{jkt}^{(q)} > 0, \\ \bar{c}_{jkt}^{(q)} & \text{if } x_{jkt}^{(q)} = 0. \end{cases}$$

Stopping condition. If two consecutive solutions in the above algorithm are equal, then the linear cost coefficients and the objective function values in the following iterations will be identical. As a result, once S(q) = S(q−1) there can be no more improvement. Therefore, a natural stopping criterion is to terminate the algorithm when S(q) − S(q−1) < ε. An alternative is to terminate the algorithm after a fixed number of iterations if the above criterion is not satisfied.

The initiation and updating schemes for fixed charge and piecewise-linear concave cases are analyzed below separately. In the remainder of this dissertation we will refer to the local search approach that uses DSSP to generate the initial solutions as LS-DSSP.

3.2.1 Fixed Charge Case

Kim and Pardalos [20] investigated two different ways to initiate the algorithm. The initiation scheme presented here is shown to provide better results when used for large scale problems (Figure 3–8). The initial value of the linear cost we use is c(jt0) =

cjt + sjt/Wjt. This is simply the LP relaxation of the original problem. However, note that the problems we are solving are NFPs which can be solved much faster than LPs.

The linear cost coefficients are updated in such a way that they reflect the variable costs and the fixed costs simultaneously in the following way:

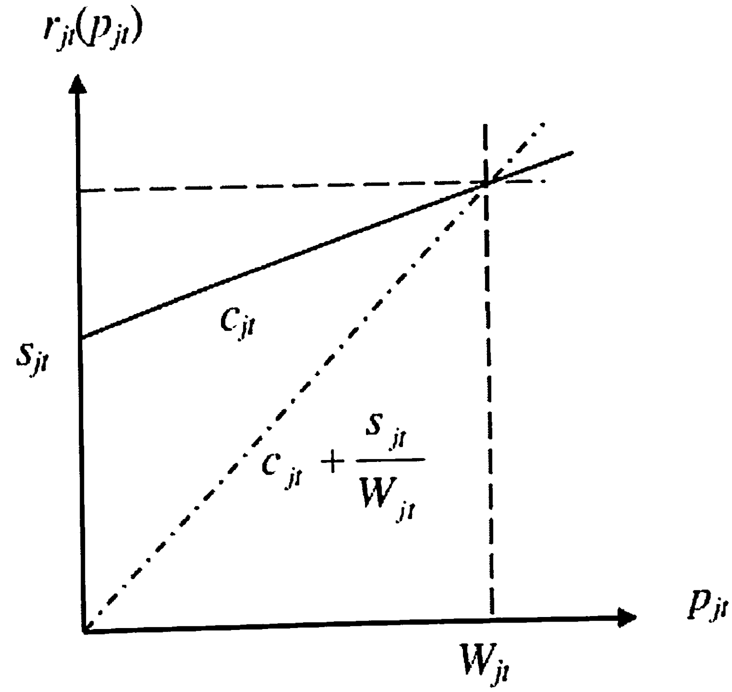

Figure 3–8: Linearization of the fixed charge cost function.

$$\bar{c}_{jt}^{(q)} = \begin{cases} c_{jt} + s_{jt}/p_{jt}^{(q)} & \text{if } p_{jt}^{(q)} > 0. \\ \bar{c}_{jt}^{(q)} & \text{if } p_{jt}^{(q)} = 0. \end{cases}$$

$$\bar{c}_{jkt}^{(q)} = \begin{cases} c_{jkt} + s_{jkt}/x_{jkt}^{(q)} & \text{if } x_{jkt}^{(q)} > 0. \\ \bar{c}_{jkt}^{(q)} & \text{if } x_{jkt}^{(q)} = 0. \end{cases}$$

3.2.2 Piecewise-linear Concave Case

When production or transportation costs are piecewise-linear and concave, the problem can be transformed into a fixed charge network flow problem as described in section 2.4.2. After the transformation, the solution approach given in section 3.3.1 can be used to solve the problem. However, the transformed problem does not always lead to generation of feasible solutions since the problem is not solved to optimality. Therefore, we apply the procedure to the original problem instead of the extended network with some modification to account for each piece in the cost functions.

The linear cost coefficients are initialized in the same way as the fixed charge case, i.e. c(jt0) = cjt + sjt/Wjt for the production arcs. To update the costs, the flow values are checked to see which interval it falls into and the corresponding fixed and variable costs are used. The updating scheme (Figure 3–9) is as follows:

$$\bar{c}_{jt}^{(q)} = \begin{cases} c_{jt} + s_{jt}/p_{jt}^{(q)} & \text{if } \beta_{jt,i-1} < p_{jt}^{(q)} \le \beta_{jt,i}. \\ \bar{c}_{jt}^{(q)} & \text{if } p_{jt}^{(q)} = 0. \end{cases}$$

$$\bar{c}_{jkt}^{(q)} = \begin{cases} c_{jkt} + s_{jkt}/x_{jkt}^{(q)} & \text{if } \beta_{jkt,i-1} < x_{jkt}^{(q)} \le \beta_{jkt,i}. \\ \bar{c}_{jkt}^{(q)} & \text{if } x_{jkt}^{(q)} = 0. \end{cases}$$

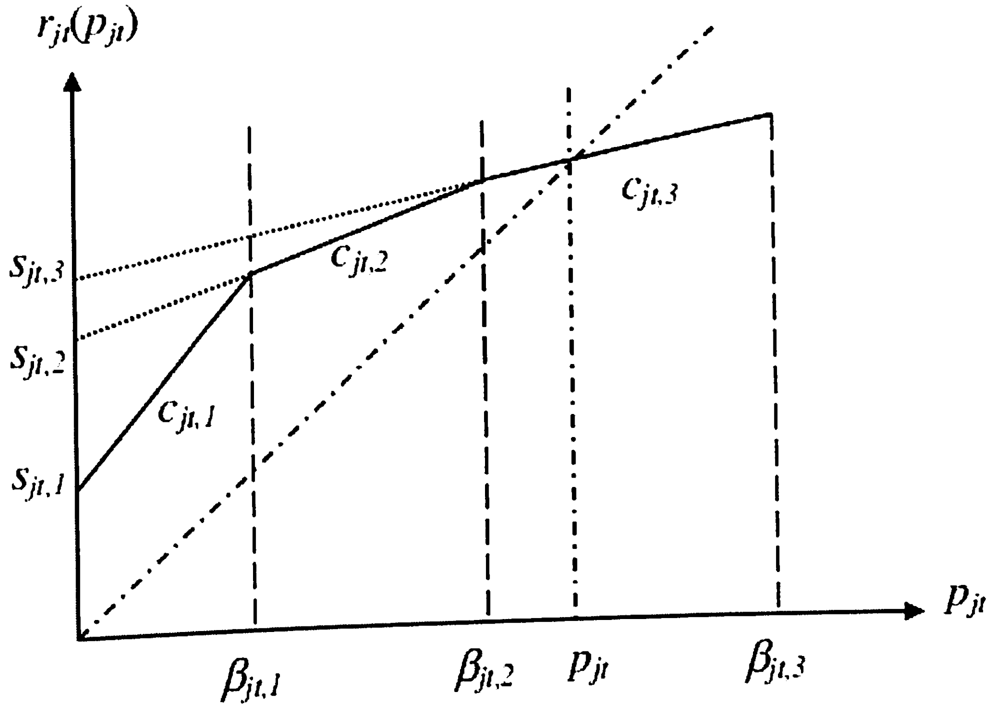

Figure 3–9: Linearization of the piecewise-linear concave cost function.

3.2.3 Performance of DSSP on Some Special Cases

In this section we analyze the performance of DSSP on some basic problems. The problems considered are simple cases with only a single time period. We identify some cases where the DSSP heuristic actually finds the optimal solution. Multiple facilities and one retailer. If there is only a single retailer but multiple facilities in the network, then the problem is easy to solve, provided that there are no capacities on the production and transportation arcs and the cost functions are concave. The problem in this case simply reduces to a shortest path problem (Figure 3–10) which can be solved in O(J) time by complete enumeration. In an optimal solution only one pair of production and transportation arcs will be used since this is an uncapacitated concave minimization problem. The optimal solution has the following characteristics (j* indicates the facility used in the optimal solution):

$$p_{j^*1} = x_{j^*11} = d_{11}$$

$$p_{j1} = x_{j11} = 0 \quad \forall\, j \in \{1, 2, \ldots, J\} \backslash \{j^*\}$$

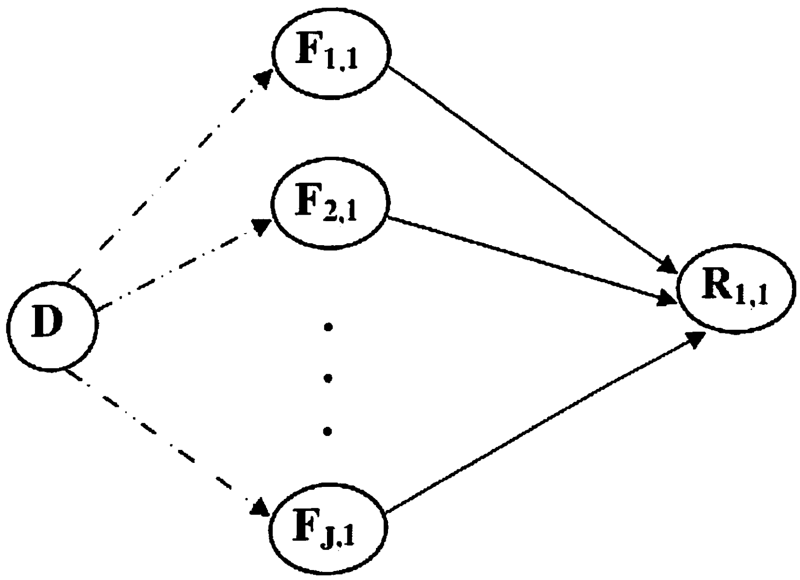

Figure 3–10: A bipartite network with many facilities and one retailer.

Proposition 3 The DSSP finds the optimal solution in J +2 iterations, in the worst case, for the single retailer problem. Proof. In every iteration of the DSSP heuristic the following network flow problem is solved (derived from problem (P1) for a single retailer): minimize

$$\sum_{j=1}^{J} (\bar{c}_{j1}^{(q-1)} + \bar{c}_{j11}^{(q-1)}) p_{j1}^{(q)}$$

$$\sum_{j=1}^{J} p_{j1}^{(q)} = d_{11}$$

$$p_{j1}^{(q)} \geq 0 \qquad j = 1, \ldots, J.$$

Although (P1) is uncapacitated, the heuristic requires an upper bound to initialize the linear cost coefficients. Therefore, let Wj1 and Uj11 be the upper bounds for the production and transportation arcs, respectively, such that. Hence, the initial cost coefficients are.

The heuristic will visit each facility at most once and facility, j*, will be visited at most three times (twice in the last two iterations and possibly once during the previous iterations). The total number of iterations, therefore, is J + 2 in the worst case. If then the optimal solution will be found in only 2 iterations. 2

Two facilities and two retailers. When there are only two facilities and two retailers the problem is obviously easy since there are only a few feasible solutions to consider (Figure 3–11). If there are no production and transportation capacities then the two retailers are not competing for capacity and they can act independently.

This indicates that the problem decomposes into two single-retailer problems both of which can be solved to optimality using the DSSP. In general, if there are J facilities, K retailers, T periods, and no capacity constraints, then the problem can be solved to optimality by solving KT single-retailer problems (with the additional condition that only distribution costs are nonlinear).

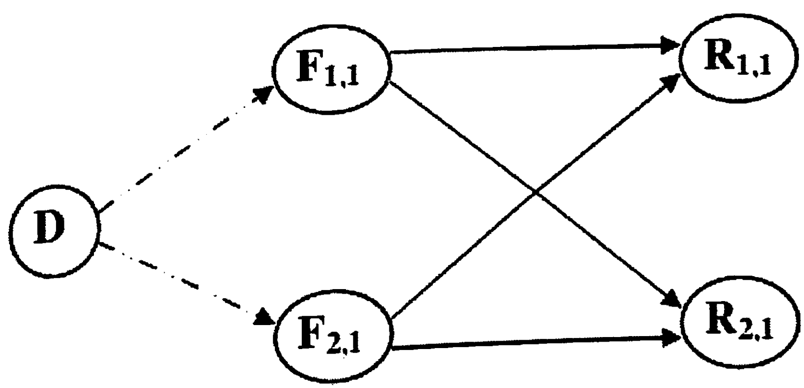

Figure 3–11: A bipartite network with two facilities and two retailers.

If, however, the facilities have production capacities then the DSSP may fail to find the optimal solution. We will illustrate this by creating an example for which DSSP fails to find the optimal solution. We assume production and distribution costs are fixed charge rather than general concave functions to simplify the analysis. The MILP formulation for the problem in Figure 3–11 with fixed charge costs can be given as (derived from problem (P2) in section 2.4.1)

Multiple facilities and multiple retailers. If the problem has multiple facilities, multiple retailers, and concave cost functions, then it falls under the category of MCCNFP which is known to be NP-hard (section 2.3).

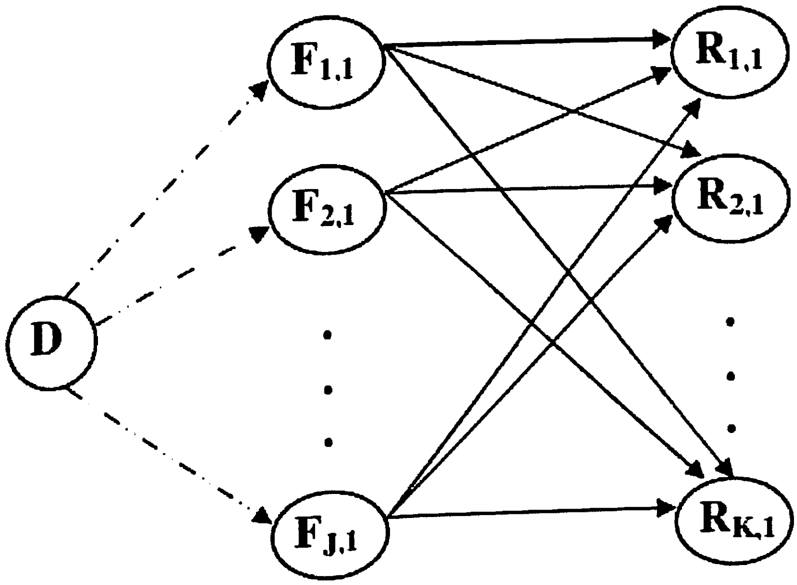

Figure 3–12: A bipartite network with multiple facilities and multiple retailers.

However, if (i) the distribution costs are concave, (ii) there are equal number of facilities and retailers, (iii) the demands at the retailers are all the same, and (iv) all production capacities are equal to the constant demand, then the problem becomes easier (Figure 3–12). In other words, if J = K and Wj1 = dk1 = d for j, k = 1, 2, ..., J the problem formulation is

$$\sum_{j=1}^{J} r_{j1}(p_{j1}) + \sum_{j=1}^{J}\sum_{k=1}^{J} f_{jk1}(x_{jk1})$$

$$
\begin{aligned}
p_{j1} - \sum_{k=1}^{J} x_{jk1} &= 0 & j &= 1, \ldots, J, \\
\sum_{j=1}^{J} x_{jk1} &= d & k &= 1, \ldots, J, \\
p_{j1} &\leq d & j &= 1, \ldots, J, \\
x_{jk1} &\leq d & j, k &= 1, \ldots, J, \\
p_{j1}, x_{jk1} &\geq 0 & j, k &= 1, \ldots, J.
\end{aligned}
$$

Due to tight production capacities the only feasible way of meeting the demand is if all facilities produce at their full capacity. This means that pj1 = d for j = 1, 2, ..., J so, the production variables can be dropped from the formulation and the problem reduces to

$$
\sum_{j=1}^{J} \sum_{k=1}^{J} f_{jk1}(x_{jk1})
$$

$$
\begin{aligned}
\sum_{k=1}^{J} x_{jk1} &= d & j &= 1, \ldots, J, \\
\sum_{j=1}^{J} x_{jk1} &= d & k &= 1, \ldots, J, \\
x_{jk1} &\geq 0 & j, k &= 1, \ldots, J.
\end{aligned}
$$

Note that constraints (3.36) and (3.37) together with zjk1 ∈ {0, 1} are the constraints of the well-known assignment problem. If the transportation costs, fjk1(·), are concave then we will refer to as the concave assignment problem.

Proposition 4 The DSSP finds the optimal solution in 2 iterations for a concave assignment problem. Proof. If DSSP is used to solve the following network flow problem is solved in every iteration:

$$\sum_{j=1}^{J}\sum_{k=1}^{J} \overline{c}_{jk1}.x_{jk1}$$

$$\begin{aligned}
\sum_{k=1}^{J} x_{jk1} &= d & j &= 1,\ldots,J. \\
\sum_{j=1}^{J} x_{jk1} &= d & k &= 1,\ldots,J. \\
x_{jk1} &\geq 0 & j,k &= 1,\ldots,J.
\end{aligned}$$

(P8)

This is equivalent to solving the LP relaxation of a linear assignment problem. Therefore, the solution to (P8) gives the optimal solution to (P7). The objective function coefficients in the next iteration of DSSP will be the same since. The same solution will be found in the second iteration and DSSP will terminate with the optimal solution.

3.3 Greedy Randomized Adaptive Search Procedure (GRASP)

The Greedy irregular adaptative Search Procedure is associate reiterative method that gives a possible resolution at each iteration for combinatorial optimisation issues. GRASP is sometimes enforced as a multi-start procedure wherever every iteration consists of a construction part and an area search part. within the construction part, a irregular greedy perform is employed to make up associate initial resolution. This resolution is then used for improvement makes an attempt within the native search part. This reiterative method is performed for a fixed range of iterations and also the final result's merely the simplest resolution found over all iterations. the quantity of iterations is one in every of the 2 parameters to be tuned. the opposite parameter is that the size of the candidate list employed in the development part.

GRASP has been applied with success for a large vary of research and trade issues like programing, routing, logic, partitioning, location, assignment, producing, transportation, telecommunications. Festa associated Resende offer an extended list of GRASP literature. GRASP has been enforced in numerous ways in which with some modifications to reinforce its performance. Here, we tend to develop a GRASP for the PID downside. within the following, we tend to discuss generation of initial solutions. we tend to offer 2 different construction procedures. The second construction procedure, known as the modified construction part, is developed to boost the performance of the heuristic procedure.

3.3.1 Construction Phase

In the construction part a possible resolution is made step by step, utilizing a number of the matter specific properties. Since our issues square measure uncapacitated cupular price network flow issues the best resolution are a tree on the network. Therefore, we have a tendency to construct solutions wherever every distributor is equipped by one facility. the development part starts by connecting one among the retailers to at least one of the facilities. The procedure finds the facilities that provide rock bottom per unit production and transportation price for a distributor, taking into consideration the effect that already connected retailers wear the answer.

The cost, θ, of assignment facility j to distributor k. the most cost effective connections for this distributor square measure then place into a restricted candidate list (RCL) and one among the facilities from RCL is chosen every which way. the scale of the RCL is one among the parameters that needs standardisation. Hart and Shogan and Feo and Resende propose a cardinality-based scheme and a value-based scheme to build an RCL. In the cardinality-based scheme, a fixed number of candidates are placed in the RCL. In the value-based scheme, all candidates with greedy function values within (100α)% of the best candidate are placed in the RCL where $\alpha \in [0, 1]$. We use the value-based scheme and a candidate facility is added to the RCL if its cost is no more than a multiple of the cheapest one. Finally, when the chosen facility is connected to the retailer, the flows on the arcs are updated accordingly. The procedure is given in Figure 3–13. When α, in Figure 3–13, is zero the procedure is totally randomized and the greedy function value, θj, is irrelevant. However, when α is equal to 1 the procedure is a pure greedy heuristic without any randomization.

Holmqvist develop a similar GRASP for single source uncapacitated MCCNFP. Their problems are different from ours because they use different cost functions and network structures. Also, our approach differs from theirs in the greedy function used to initialize the RCL. They use the actual cost where as we check the unit cost of connecting retailers to one of the facilities. Due to the presence of fixed costs the unit-cost approach performed better. We have tested both approaches and our approach consistently gave better results for the problems we have tested. The results are presented.

The construction procedure in Figure 3–13 handles every amount severally. In different words, the initial answer created doesn't carry any inventory and demands ar met through production inside identical amount. A second approach is to permit demand to be met by production from previous periods. The second approach junction rectifier to raised initial solutions, however the final solutions when native search we have a tendency tore worse compared to those we got from the first approach. One rationalization to the current is that the initial solutions within the second approach ar probably already native optima. within the first approach, however, we have a tendency to don't begin with a really sensible answer in most cases however the native search section improves the answer. The native search approach that uses the development procedure represented during this section are going to be mentioned as LS-GRASP from here on

```
procedure Construct
  for (t = 1, ..., T) do
    p_jt := 0, j = 1, ..., J
    x_jkt := 0, j = 1, ..., J, k = 1, ..., K
    for (k = 1, ..., K) do
      θ_j := f_jkt(d_kt)/d_kt + r_jt(p_jt + d_kt)/(p_jt + d_kt), j = 1, ..., J
      Θ := min{θ_j : j = 1, ..., J}
      RCL := {j : θ_j ≤ Θ/α, 0 ≤ α ≤ 1}
      Select l at random from RCL
      p_lt := p_lt + d_kt
      x_lkt := d_kt
    end for k
  end for t
  return the current solution, S
end procedure
```

Figure 3–13: The construction procedure.

3.3.2 Modified Construction Phase

In a multi-start procedure, if initial solutions are uniformly generated from the feasible region, then the procedure will eventually find a global optimal solution (provided that it is started enough times). If initial solutions are generated using a greedy function, they may not be in the region of attraction of a global optimum because the amount of variability in these solutions is typically smaller. Thus, the local search will terminate with a suboptimal solution in most cases. GRASP tries to find a balance between these two extremes by generating solutions using a greedy function and at the same time adding some variability to the process. After the first set of results we obtained by using the construction procedure explained in section 3.4.1, we wanted to modify our approach to diversify the initial solutions and increase their variability.

In order to diversify the initial solutions we use the same greedy function but we allow candidates with worse greedy values to enter the RCL. The modified construction procedure is similar to the procedure given in Figure 3–13. The only difference is in the way the RCL is constructed. The RCL in the modified construction procedure is defined as

$$\text{RCL} := \{j : \underline{\Theta} \leq \theta_j < \underline{\Theta} + \Delta_{\Theta}\}$$

3.4 Lower Bounds

The local search procedures presented above give suboptimal but feasible solutions to the PID problem. These feasible solutions are upper bounds to our minimization problem. To evaluate the quality of the heuristics, when the optimal solution value is unavailable, we need a lower bound. It is as important to get good lower bounds as it is to get good upper bounds. For the fixed charge case, the LP relaxation of (P2), unfortunately, gives poor lower bounds, particularly for problems with large fixed charges. Therefore, we reformulated the problem, which led to an increase in problem size but the LP relaxation of the new formulation gave tighter lower bounds. For the piecewise-linear concave case we reported lower bounds obtained from CPLEX in Chapter 4. The MILP formulation is run for a certain amount of time and CPLEX may fail to find even a feasible solution by that time, but a lower bound is available. For problems with piecewise-linear and concave costs, we have also obtained lower bounds using the procedure given below. These problems were first transformed into fixed charge network flow problems using the node separation procedure (section 2.4.2). Once they are fixed charge type problems the procedure provided below can be used to find lower bounds.

To reformulate the problem with fixed charge costs we define new variables which help us decompose the amount transported from a facility to a retailer by origin. Figure 3–14 gives a network representation for the new formulation.

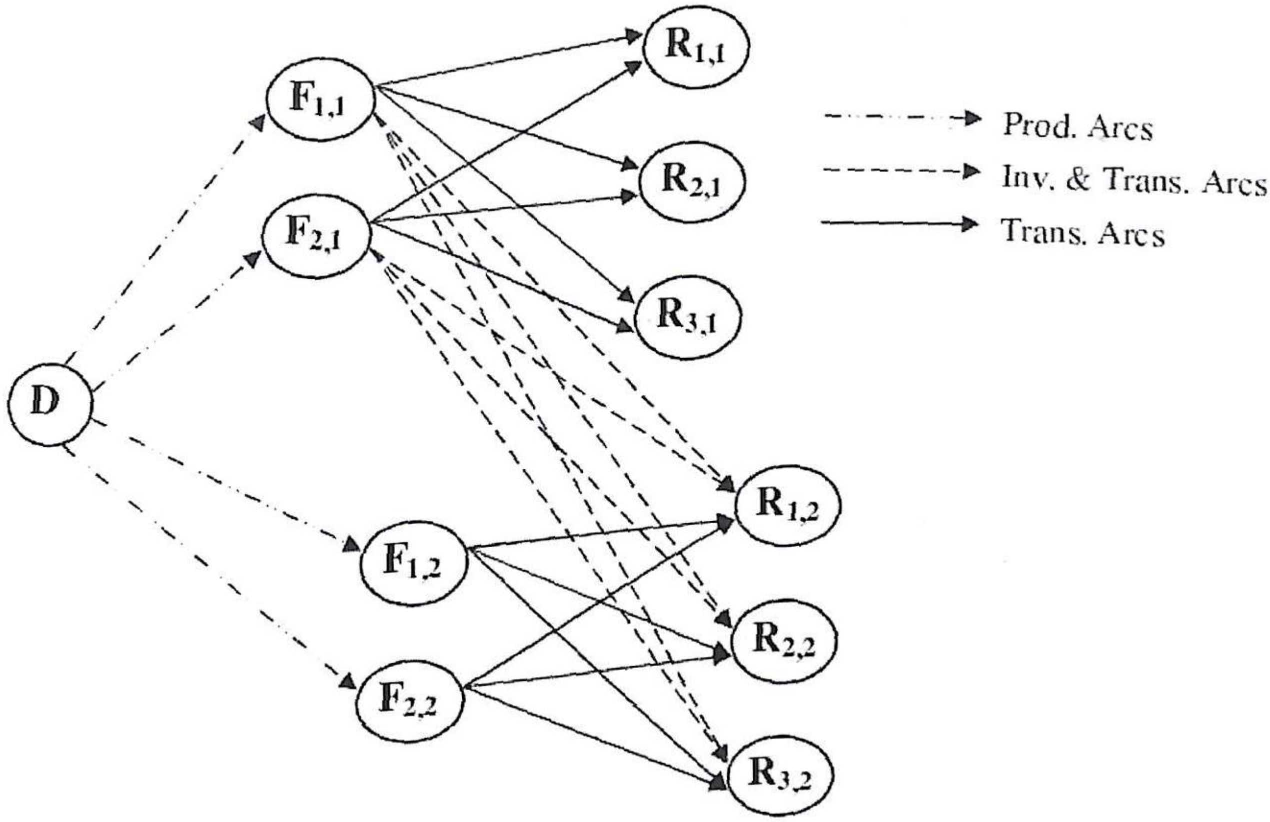

Figure 3–14: Extended network representation.

carried at facility j from period t to t+1 is equal to the total demand of all retailers in periods t + 1 to T satisfied by facility j through production in periods 1 to t. Finally, equation (3.44) indicates that the amount of production at facility j during period t is equal to the amount of demand met from this production for all retailers in periods t through T. Substituting (3.42), (3.43) and (3.44) into (P2) will give the following new formulation:

$$\sum_{j=1}^{J}\sum_{t=1}^{T} s_{jt}y_{jt} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T} s_{jkt}y_{jkt} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{\tau=1}^{t} \nu_{j\tau kt}y_{j\tau kt}$$

Proposition 5 For uncapacitated PID problems with fixed charge costs the optimal cost of the LP relaxation of (P10) is greater than or equal to the optimal cost of the LP relaxation of (P2).

Proof. Let (yjt, yjτkt) be a feasible solution to (P10). Multiplying the constraints by dkt and summing them over k and t leads to (3.55). Constraints can be reached in a similar way. Substituting the yjτkt values into, and (3.44) will give a feasible solution to (P2). It follows that every solution to the LP relaxation of (P10) gives rise to a feasible solution of the LP relaxation of (P2).

4.1 Existing Models and Algorithms

Now exists a lot of models, methodologies and algorithms for solving optimization problems. First we made research for this models and algorithms. We try to get cons and pros to analyze and best features for featured system. Existing algorithms

High Dencity Routing - points to waste of collection, local post deliveries, snow cleaning, newsletter delivering and meter reading actions. High Density Routing is used when you have big number of locations to visit on the same scheduled time.

Point-to-Point Routing – refers to the sales, deliveries and collections with points to visit less than 200 for each day. With this algorithm, boundary is not so important as the schedule and sequence.

Paired Routing – points to para transit, car with armor routing and etc. This algorithm is similar to point-to-point routing algorithm, but the sequence requires to every stop to be "paired" with trip.

First of all, we need to find mathematical calculations and formulas for route optimization. Route optimization algorithms are the mathematical calculations and formulas to solve the routing formulas which includes traffics and having many different routes and points.

Types of routing:

- Vehicle Routing Problem
- Travelling Salesman Problem
- Ant Colony Problem and etc

For proposed algorithm we should use Vehicle Routing Problem. Because this solution is more close to problems solution. And for algorithm we chose existing engine for which give us mathematical result for future using in algorithm.

4.2 Proposed Algorithm

Many existing applications of different routing problems, a wide variety of researchers and programmers have focused on developing solutions to them. See previous parts. This research described above has shown that existing algorithms based on accurate and heuristic algorithms aims to achieve the lowest transportation cost possible. To our knowledge, no study has considered the followings:

- Time window
- Step by step planning
- Using real path instead of straight route
- Ruin and recreate procedure
- Constraint programming

This features constraints transportation expenses in logistics enterprises, and so it is necessary for logistics to take the time window constraints into consideration. Next parts explain why wee need add this features and how it helps us to make optimal algorithm for logistics.

4.2.1 Time window

This paper presents a transportation cost model with following constraints, which is modeled through modified genetic algorithm. In our research, the simultaneous minimization of total route and total transportation cost are considered following functions. The model is formulated under the following assumptions:

1. Time window are soft, and the time windows specified by customers are elastic.
2. The service time for a vehicle at its destination is equal to zero.
3. A route is defined as starting from a warehouse, going through a number of customers, and ending at the depot.
4. Every customer on the route must be visited only once by one of the courier or vehicles.

The transportation cost model in this paper takes time window constraints into consideration. Any customer i must be serviced within a pre-defined time interval [ES, LF], limited by an earliest arrival time ES, and a latest arrival time LF. The vehicles arriving time later than the latest arrival time are penalized, while those arriving time earlier than the earliest arrival time have to pay for the inventory cost involved.

In order to formulate the model, the notations in it are defined as follows (Figure 4.1 and 4.2):

BEGIN:

(1) Randomly generate a number q_{cross}, such that $1 \le q_{cross} \le K$ and $\beta^F_{cross} \neq \varnothing$;
(2) Let $I^D = I^M \setminus \{\beta^F_{cross}\}$;
(3) Randomly select a gene g^F in β^F_{cross};
(4) **FOR** i = 1 TO length(I^D);
(5) Generate the partial schedule $P_i = (I^D_{1\to i-1}, g^F, I^D_{i\to end})$;
(6) Compute the total transportation cost C_i of P_i. If P_i is infeasible, let C_i = a sufficiently large number;
(7) Let $i^* = \min_i\{r \mid C_r\}$ and reinsert gene g^F in the position i^* of I^D;
(8) Update $I^D = (I^D_{1\to i^*-1}, g^F, I^D_{i^*\to end})$ and $\beta^F_{cross} = \beta^F_{cross} \setminus g^F$; **ENDFOR**
(9) If $\beta^F_{cross} \neq \varnothing$, return to Step (3); otherwise, terminate the procedure.

END

Figure 4.1 Procedure 1

BEGIN

Input: all necessary data related to customers and vehicles
GA parameters: POP, p_{cross}, p_{mut}
Generate a new population of individuals
While maximal iteration number is not met DO
Evaluate the individuals
Select the better individuals
Generate the offspring population using evolutionary operators
Output: a set of non-dominated solutions

END

Figure 4.2 Procedure 2

4.2.2 Real path route and of straight route

In the real world, vehicles in a Vehicle Routing Problem have to follow the roads. They cannot travel in a directly line from customer to customer. Most Vehicle Routing Problem research papers ignore this implementation detail. Using road distances (not straight distances) does not impact the Vehicle Routing Problem too much but it does result in a few extra challenges.

First off all, we will need real datasets of points. Unfortunately, open Vehicle Routing Problem datasets with road distances are less than needed in the VRP community. The Vehicle Routing Problem database with dataset with 29 locations for testing. So we generate some data by myself with requirements:

- I need to use Google or Yandex Maps for real distance roads in KM between every pair of point in the dataset. For example, use highways when reasonable over small roads.
- To compare every dataset, I will generate straight and road distance.
- Vehicle Routing Problem compare scalability, I need to generate a similar dataset in many orders of magnitude.
- Add reasonable vehicle parameters and customer demands, for the vehicle capacity constraint in VRP.

For clear vision, I will focus on choosen dataset of points, which has 50 locations, 10 vehicles with capacity 125 each. Using straight distance (which calculates the euclidean distance based on latitude and longitude) results in figure: 4.3

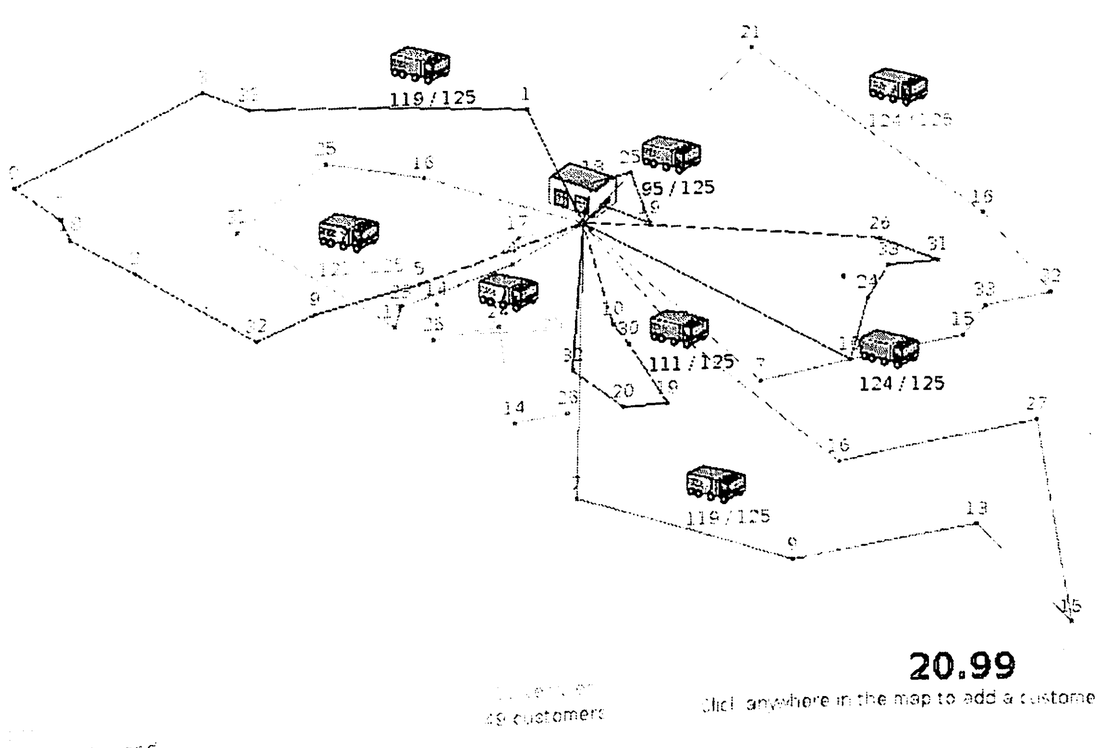

Figure 4.3 Straight distance

The total distance is 20.99 KM, it doesn't say any interesting information because, our vehicles fly between points. Next thing what I do is, apply real distance solution on my straight roads which is shown in figure 4.4 The road distance is 2 366.76 KM. Now I will compare the road distance which I generated using Graphhopper and OpenStreetMap. In the figure 4.5 we can see that road distance is 108.45 KM less than air distance, so it's almost 5% better.

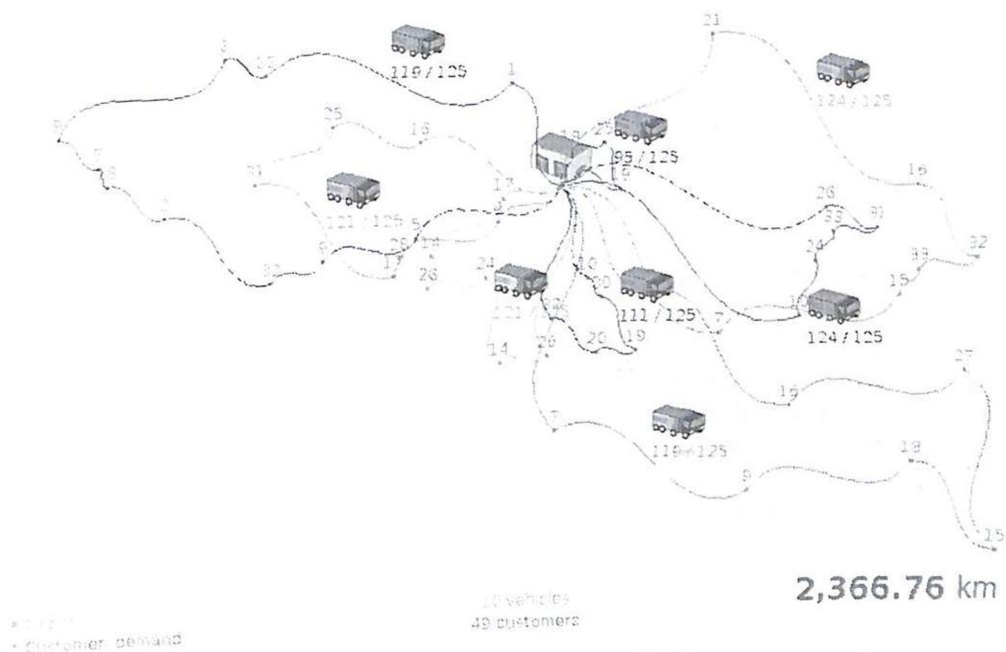

Figure 4.4 Real path road distance

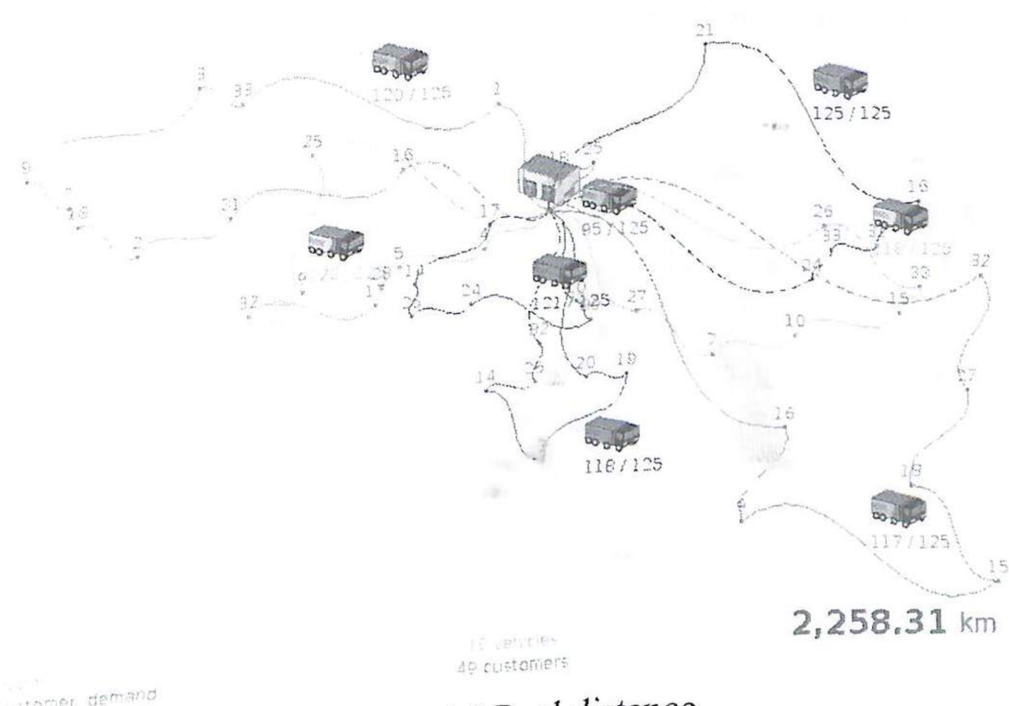

Figure 4.5 Real distance

4.2.3 Ruin and recreate procedure

The algorithms Ruin & Recreate is main part of the algorithm. Firstly, algorithms create the not the best routing, then I will need to apply ruin and recreate procedure on my routing to improve delivering process. The main idea of ruin and recreate is ruining route created by first part and improving it.

In first graph of Figure 4.6 we can see that it is not the optimal road to deliver. Ruin & Recreate destroys some edges between points and searches for optimal vertical for this route. In the end of ruining and recreating we will get optimal routing plan than visits all vertices with less resources.

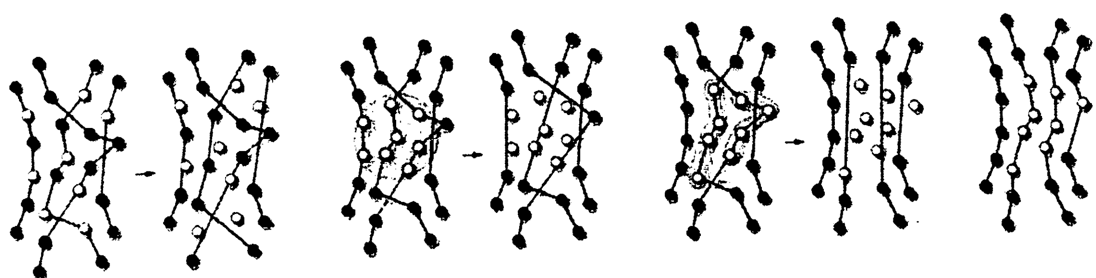

Figure 4.6 Simple Ruin & Recreate

One of the useful method of using ruin and recreate is radiusing the ruin area. When ruining algorithms looks for vertices to destroy, it takes only edges which is in selected radius, it is useful when company has geo zones in routing. View in Figure 4.7

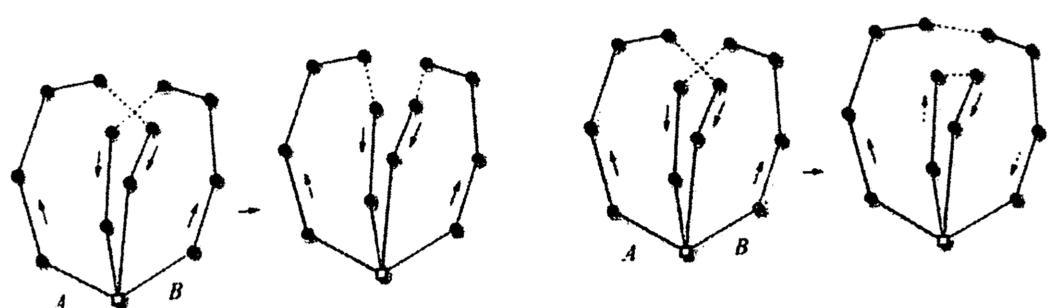

Figure 4.7 Radius Ruin

4.2.4 Constraint programming

The constraint programming is one of the way of modeling and solving combinatorial optimization problems, it includes artificial intelligence, logical programming, and operations research strategies. The benefits of Constrait programming:

- Systematic search
- Reasoning on individual at each search state
- Reduce search space filtering at variable domains
- Domains contain holes very suitable for highly combinatorial problems

We can see how ruin and recreate works. The idea of using constrait programming is applying own constraint as logistic company, which includes:

- Goods weight and volume
- Time to unload
- Maximum weight for courier's transport
- Maximum volume of goods
- Courier's working hours
- Maximum orders which courier can take for each route
- Incompatible orders
- Geozones by routes where courier works
- Geozones by prices
- Time window

COMPUTATIONAL EXPERIMENTS AND COMPARISON

5.1 Experimental Data

The primary goal of our computational experiments is to evaluate the solution procedures developed and presented in Chapter 3. For a procedure that finds an optimal solution, an important characteristic is the computational effort needed. For a heuristic procedure, one is often interested in evaluating how close the solution value

```
GRASP for Production-Inventory-Transportation Problem

CPU time                          0.00
Number of plants                     2
Number of retailers                  2
Number of periods                    2
Number of iterations                 8
Initial seed                         1

Minimum cost before local search    1288.10
Minimum cost after local search     1288.10

Production    Period       Plant  Retailer
     55.61         1           2
     38.58         2           2
  Inventory
  Shipment
     19.98         1           2         1
     35.63         1           2         2
      8.67         2           2         1
     29.91         2           2         2
All other variables are zero.
```

Figure 4–3: Sample output

is to the optimal value. One method for evaluating these characteristics is running experiments on real-world or library test problems. Real-world problems are useful because they can accurately assess the practical usefulness of a solution procedure. Library test problems are important because the properties of the problems and the performance of other procedures are usually known for these problems. However, the availability of data for either of these types of problem instances is limited. Moreover, there may also be disadvantages of using real-world and library test problems, which are briefly described later in section 4.5. Therefore, many research studies use randomly generated problem instances as we do in this dissertation. To test the behavior of the solution procedures developed for the PID problem we primarily use randomly generated problems and some library test problems from the literature. Unfortunately, we were not able to find library test problems that fit our problem requirements and characteristics exactly. Therefore, we found other library problems such as facility location problems and formulated them as PID problems. Once they were formulated as PID problems, we tested the algorithms on these problems.

5.2 Randomly Generated Problems

The behavior and performance of solution procedures are frequently tested on a collection of problem instances. The conclusions drawn about these performance characteristics strongly depend on the set of problem instances chosen. Therefore, it is important that the test problems have certain characteristics to ensure the credibility of the conclusions. Hall and Posner gave suggestions on generating experimental data which are applicable to most classes of deterministic optimization problems. They developed specific proposals for the generation of several types of scheduling data. They proposed variety, practical relevance, invariance, regularity, describability, efficiency, and parsimony as some of the principles that should be considered when generating test problems. We take these properties into account when generating our data.

The first set of issues we have a tendency to resolved were inflammatory disease issues with fixed costs production and distribution costs. LS-DSSP and LS-GRASP were accustomed resolved these issues. for a few of the issues LSM-GRASP was used. drawback size, structure, and also the price values appear to have an effect on the problem of the issues. Therefore, we have a tendency to generated 5 teams of issues. cluster one and cluster a pair of issues ar single amount issues while not inventory and teams three, 4, and five ar multi-period issues. To capture the result of the structure of the network on the problem of the issues we have a tendency to varied the amount of facilities, retailers, and periods inside every cluster. the matter sizes ar given in Table 4–2. issues one through five ar in cluster one, issues half dozen to ten ar in cluster a pair of, issues eleven to fifteen ar in cluster three, issues sixteen to twenty ar in cluster four, and cluster five consists of issues twenty one to twenty five. a complete of 1250 issues were resolved. it's been shown that the quantitative relation of the full variable price to the full fixed charge is a vital live of the matter.

Therefore, we randomly generated demands, fixed charges, and variable costs for each test problem from different uniform distributions as given in Table 4–3. Note that only the fixed charges are varied. For example, data set A corresponds to those problems that have small fixed charges. The fixed charges increase as we go from data set A to data set E. The distances between the facilities and the retailers are used as the unit transportation costs (cjkt). For each facility and retailer we uniformly generated x and y coordinates from a 10 by 10 grid and calculated the distances. The values of the parameter fi, which is used in the construction phase of LS-GRASP, were chosen from the interval (0, 1). Experiments indicated that better results were obtained for fi in the interval [0.80, 0.85].

For each problem size and for each data set we randomly generated 10 problem instances. The algorithms are programmed in the JAVA language, and CPLEX 7.0 callable libraries are used to solve the linear NFPs and the MILPs. The algorithms are compiled and executed on an IBM computer with 2 Power3 PC processors, 200 Mhz CPUs each.

Group	Problem	No. of Facilities	No. of Retailers	No. of Periods	No. of Arcs
	1	25	400	1	10,025
	2	50	400	1	20,050
1	3	75	400	1	30,075
	4	100	400	1	40,100
	5	125	400	1	50,125
	6	125	200	1	25,125
	7	125	250	1	31,375
2	8	125	300	1	37,625
	9	125	350	1	43,875
	10	125	400	1	50,125
	11	10	70	20	14,390
	12	15	70	20	21,585
3	13	20	70	20	28,780
	14	25	70	20	35,975
	15	30	70	20	43,170
	16	30	60	5	9,270
	17	30	60	10	18,570
4	18	30	60	15	27,870
	19	30	60	20	37,170
	20	30	60	25	46,470
	21	30	50	20	31,170
	22	30	55	20	34,170
5	23	30	60	20	37,170
	24	30	65	20	40,170
	25	30	70	20	43,170

Table 4–2: Problem sizes for fixed charge networks

The errors found for issues exploitation information set A were the tiniest. this is often expected since the mounted charges for this set ar little. because the fixed cost will increase, the linear approximation isn't a detailed approximation any longer. Thus, the error bounds for information set E were the best. even so, all errors reportable were less. The maximum error encountered for data set A set was 0:02% for both heuristics. The results indicate that LS-DSSP and LS-GRASP are both powerful heuristics for these problems, but another reason for getting such good solutions is because the problems in set A are relatively easy. We were actually able to find optimal solutions from CPLEX for all problems in this set in a reasonable amount of time. On average, LS-DSSP took less than a second to solve the largest problems in Groups 1 and 2. LS-GRASP with 8 iterations took about 1.5 seconds for the same problems. Finding lower bounds also took about a second, whereas it took more than 25 seconds to find optimal solutions for each of these problems using CPLEX. The problems in Groups 3, 4, and 5 were relatively more difficult. Average times were 8, 10, 15, and 50 seconds for LS-DSSP, LS-GRASP with 8 iterations, LP, and CPLEX, respectively. Optimal solutions were found for all problems using data set B as well. However, for data set C, optimal solutions were found for only Group 1, 2, and 3 problems and for some of the problems in Groups 4, and 5. For data set D, optimal solutions were found for some of the Group 1 and 2 problems. Finally, data set E was the most difficult set. CPLEX was able to provide optimal solutions for only a few of the Group 1 and 2 problems and none for the other 3 groups. From Figure 4–4 we can see that LS-DSSP gave better results for Group 1 and 2 problems and LS-GRASP gave slightly better results for problems in Groups 3, 4, and 5. Similar behavior was observed for the other data sets.

The results also indicate that LS-GRASP was more robust in the sense that the quality of the solutions was not affected greatly by the network structure. The proposed local search also proved to be powerful. For example, average errors for the problems in Figure 4–4-d without the local search were roughly 2:9% for the DSSP and 11% for GRASP with 8 iterations. If more iterations are performed, LS-GRASP finds better solutions. Figure 4–5 shows that the errors decrease as the number of iterations increase. However, the computational time required by LS-GRASP increases linearly as more iterations.

5.3 Library Test Problems

As we mentioned in section 4.2, real-world problems and library test problems are useful, but it is usually difficult to obtain these kind of problems. He also notes that library data sets may quickly become obsolete.

Moreover, it is difficult to generalize results from a small set of problem instances. Another disadvantage of using library problems is that researchers may be tempted to tune their algorithm so that it works well on that particular set of instances. This may favor procedures that work poorly for general problems and may disfavor those procedures that have superior performance. We could not find any PID problems in the OR-library, but there were uncapacitated warehouse location problems. We formulated these as PID problems and used LS-DSSP and LS-GRASP to solve them. The data files have the following information:

Problem	No. of Facilities	No. of Retailers	No. of Periods	No. of Pieces	No. of Arcs (org)	No. of Arcs (ext)
40	125	200	1	8	25,125	201.000
43	10	70	20	8	14,390	1.177.600
46	30	60	5	8	9.270	217.200
49	30	50	20	8	31.170	2.524.800

Table 4–17: Problem sizes for the extended formulation after NSP

- number of potential warehouse locations,
- number of customers,
- the fixed cost of opening a warehouse,
- the demand of each customer, and
- the total cost of flowing all of the demand from a warehouse to a customer.

The warehouses in the location problems are treated as the facilities in the PID problem and the customers are treated as the retailers. There is only a single period and the fixed costs of opening a warehouse are used as the fixed production costs at the facilities. Note that the variable costs of production will be zero.

Data Set	Problem	Error (%) LS-DSSP	Error (%) LSM-GRASP	Error (%) CPLEX	LP-NSP CPU (seconds)
A	40	0.88	0.71	1.81	6.761
	46	0.77	0.66	2.73	1,370
B	40	1.04	0.72	27.67	6,340
	46	0.78	0.48	3.69	1.662
C	40	1.17	0.51	N/A	6,053
	46	1.43	0.73	6.59	1,617
D	40	1.89	0.73	N/A	6,079
	46	1.48	1.39	N/A	2.350
E	40	1.65	1.41	N/A	7,126
	46	2.94	2.66	N/A	3,626

Table 4–18: Summary of results for problems 40 and 46 after NSP.

The distribution 85 arcs, on the other hand, will have linear cost functions since there is no fixed cost of placing an order in the location problems.

A general purpose solver, CPLEX, was used in an attempt to solve our problems optimally. We compared the solution values obtained from LS-DSSP, LS-GRASP, and LSM-GRASP to the solution values obtained from CPLEX. Due to the difficulty of the problems CPLEX failed to find an optimal solution in most cases. In fact, for some problems CPLEX was not able to find even a feasible solution. We initially compared our solution values to lower bounds obtained from CPLEX. However, these lower bounds were poor, particularly for problems with high fixed charges and for problems with piecewise-linear and concave costs. Therefore, we provided extended formulations for these difficult problems and solved the LP relaxations of these extended formulations. A disadvantage of these formulations is that the problems grow in size and they become harder to solve. However, the lower bounds obtained from the extended formulation were much tighter compared to the ones obtained from CPLEX.

SUMMARY

We have developed local search algorithms for the incapacitated concave production-inventory-distribution problem. We first characterized the properties of an optimal solution to incapacitated PID problems with concave costs, and then presented DSSP and GRASP algorithms which take advantage of these properties. DSSP and GRASP provide initial solutions for the local search procedure. Computational results for test problems of varying size, structure, and cost characteristics are provided. The first set of problems for which experimental results were presented was incapacitated PID problems with linear inventory costs, and fixed charge production and distribution costs. The second set of problems included PID problems with fixed charge production costs, and linear inventory and distribution costs. These problems had production capacities. Since our local search algorithms are designed for incapacitated problems, only DSSP was used to solve these problems because DSSP works for both capacitated and incapacitated problems. The third set of computational results presented were for incapacitated PID problems with piecewise-linear and concave production costs, fixed charge distribution costs, and linear inventory costs.

A general purpose solver, CPLEX, was used in an attempt to solve our problems optimally. We compared the solution values obtained from LS-DSSP, LS-GRASP, and LSM-GRASP to the solution values obtained from CPLEX. Due to the difficulty of the problems CPLEX failed to find an optimal solution in most cases. In fact, for some problems CPLEX was not able to find even a feasible solution. We initially compared our solution values to lower bounds obtained from CPLEX. However, these lower bounds were poor, particularly for problems with high fixed charges and for problems with piecewise-linear and concave costs. Therefore, we provided extended formulations for these difficult problems and solved the LP relaxations of these extended formulations. A disadvantage of these formulations is that the problems grow in size and they become harder to solve. However, the lower bounds obtained from the extended formulation were much tighter compared to the ones obtained from CPLEX.

The computational experiments indicated that LS-DSSP was more sensitive to problem structure and input data, but the errors obtained from LS-GRASP and LSM-

GRASP were more robust. LS-DSSP gave smaller errors for problems with fixed charge costs, particularly for single period problems. LSM-GRASP, in general,gave better results for problems with piecewise-linear and concave costs. However, the CPU times required by LS-DSSP were much smaller compared to the CPU times of LS-GRASP and LSM-GRASP. LS-DSSP has a natural stopping condition, but LS-GRASP and LSM-GRASP can be performed for many iterations. Thus, the GRASP approaches lead to better results if more iterations are performed in the expense of extra computational time. The CPU times of the local search procedures with both DSSP and GRASP can further be improved since they are multi-start approaches and are suitable for parallel implementation.

The main contributions of this work can be summarized as follows:

- Several local procedures are developed that provide solutions to uncapacitated PID problems with concave costs in a reasonable amount of time.
- An extended formulation is given for PID problems with fixed charges costs. It is also shown that the LP relaxation of the extended formulation gives lower bounds that are no worse than the lower bounds obtained from the LP relaxation of the original formulation.
- Extensive amount of computational results are presented that show the effectiveness of the proposed solution approaches.
- Some special PID problems are identified for which one of the heuristic procedures finds an optimal solution.
- The codes developed for the implementation of the algorithms are made available for distribution.

The proposed solution approaches are useful in practice, particularly, for production, inventory, and distribution problems for which the supply chain network is large. For example, a network with 30 facilities, 70 retailers, 10 time periods, and fixed charge costs is quite big. Finding optimal solutions for networks of this size, and even for smaller networks, may be computationally expensive or impossible. The computational results indicated that if the number of retailers is much larger.

than the number of facilities, the heuristic approaches performed better. Therefore, we recommend the use of the proposed algorithms for problems that can be modeled as PID problems which have the characteristics mentioned above.

CONCLUSION

Today's market condition becomes additional volatile and causes additional pressure on value and speed. thanks to high competition within the economic process market and also the additional hard to please client, the merchandise life cycle and time to plug become shorter, price cutting war is harder, and also the responsiveness to dynamical demand is additional crucial. Beside high product quality, general value and latency appear to be the key success factors and vital missions of firms and its provide chain and supply functions got to optimize so as to realize the value and supplier leadership.

In addition, thanks to this market condition increasing the importance of four major supply performances that ar the delivery time, delivery value, delivery quality, and delivery utilization; the provision chain and supply functions have to be compelled to attain all performances with the minimum total value. the choice choice method which mixes the whole supply prices, performance measuring, and systems improvement conception altogether is made to make sure the optimum deciding within the supply operations. The conversion of some performance measurements like the delivery time and delay time into penalties is generated. The new approach are expected to administer the ensured optimum resolution and increase the transparency of the choice creating if it's used because the SOP.

Generally, the improvement algorithmic rule during this paper ar developed with the intention to optimize the supply value together with the intangible prices. Desired objectives were thought of and reworked into prices for comparison and {every one} connected knowledge from every operate within the supply has been shared to form the method alignment and gain higher communications that ar the vital characteristics of the responsive provide chain.

REFERENCES

[1] Goetschalckx, M. & Ashayeri, J. 1989. Classification and design of order picking. Logistics World. Vol. 2, issue 2, pp. 99-106

[2] G. Eason, B. Noble, and I.N. Sneddon, "On certain functions of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

[3] J. Clerk Maxwell, A Treatise on Android, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[4] I.S. Jacobs and C.P. Bean, "Fine particles" , vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963

[5] Zeng, A.Z. & Rossetti, C. 2003. Developing a framework for evaluating the logistics costs in global sourcing processes: An implementation and insights. International Journal of Physical Distribution & Logistics Management. Vol. 33, issue 9, pp. 785-803.

[6] Özkaya, E., Joseph, V.R., Keskinocak, P. & Weight, R. 2010. Estimating and benchmarking Less-than-Truckload market rates. Transportation Research, Part E: Logistics and Transportation Review. Vol. 46, issue 5, pp. 667-682.

[7] E.H.L Aarts and J.K. Lenstra, editors. Local Search in combinatorial optimization. John Wiley & Sons, Chichester, England, 1997.

[8] A. Aggarwal and J.K. Park. Improved algorithms for economic lot size problems. Operations Research, 41(3):549–571, 1993

[9] S.J. Chung and K.G. Murty. Polynomially bounded ellipsoid algorithms for convex quadratic programming. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, Nonlinear programming, pages 439–485. Academic Press, New York, 1981

[10] M.A. Cohen and A. Huchzermeier. Global supply chain management: A survey of research and applications. In S. Tayur, R. Ganeshan, and M. Magazine, editors, Quantitative models for supply chain management. Kluwer Academic Publisher, Boston, 1999

[11] B. Ek¸sio˘glu. Global supply chain models. In C.A. Floudas and P.M. Pardalos, editors, Encyclopedia of optimization, volume 4, pages 350–353. Kluwer Academic Publisher, Dordrecht, The Netherlands, 2001

[12] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. Journal of Global Optimization, 6(2):109–133, 1995

[13] C.A. Floudas. Deterministic global optimization: Theory, algorithms and applications. Kluwer Academic Publisher, Dordrecht, The Netherlands, 2000

[14] R. Freling, H.E. Romeijn, D. Romero Morales, and A.P.M. Wagelmans. A branch-and-price algorithm for the multi-period single-sourcing problem. Technical Report 99–12, Department of Industrial and Systems Engineering, University of Florida, 1999

[15] R. Freling, H.E. Romeijn, D. Romero Morales, and A.P.M. Wagelmans. A branch-and-price algorithm for the multi-period single-sourcing problem. Technical Report 99–12, Department of Industrial and Systems Engineering, University of Florida, 1999

[16] R. Horst, P.M. Pardalos, and N.V. Thoai. Introduction to global optimization. Kluwer Academic Publisher, Dordrecht, The Netherlands, 1995