A Multi-Exchange Heuristic to Production Line Design under Free Trade Agreement

Yunsong Guo[1], Yanzhi Li[2], Andrew Lim[2] and Brian Rodrigues[3]

[1] School of Computing, National University of Singapore isc10469@nus.edu.sg

[2]Department of IEEM, Hong Kong University of Science and Technology,
Clear Water Bay, Hong Kong {ieyanzhi,alim}@ust.hk

[3]Lee Kong Chian School of Business, Singapore Management University,
50 Stamford Road, Singapore 178899 br@smu.edu.sg

Abstract

Production line design(PLD) can be viewed as a facility location problem in global supply chain. However, New cost factors such as tariff are introduced under free trade agreements which appeared about ten years ago and is becoming pervasive in global trading. A multinational manufacturer has factories all over the world to produce a product. When a demand arises from a new market, the manufacturer can fully utilize the existing resources and configure a most cost-effective production line to satisfy the new demand. Such a problem turns out to be NP-hard. We provide an integer programming model, which is hard to solve optimally while able to provide a lower bound. We then propose a multi-exchange heuristic embedded in simulated annealing to tackle the problem. Extensive experiments are conducted and comparisons are made with the lower bound and show that our algorithm performs very well.

1 Introduction

One of the most important aspects of logistics is to decide where to locate facilities. Thus facility location has attracted lots of researchers and encompassed rich results in the past several decades. A traditional facility location problem usually considers costs including facility setup cost, unit service cost such as transportation cost, production cost, etc, see [5] for a thorough summary. While with the development of industrial globalization, more cost factors have come into our view and started to play more and more important roles in cost evaluation so that they must be incorporated into the traditional facility location model to make it accurately reflect the practice. One such example is tariff issue in global manufacturing/supply chain.

With the prosperity of global supply chain, a multinational company must consider all its behaviors from raw material procurement to production planning to marketing with a globalized view in the more and more competitive environment in order to lower total cost by taking advantage of different cheaper resources in various countries. However, each government would put restrictions such as quota, tariff to limit imports so as to protect domestic markets. This would eliminate or at least weaken the cost advantage of global sourcing manufacturers. The barriers from these restrictions make obstacles for efficient global resource utilization. Free trade agreement(FTA) is an endeavor made by governments to reduce tariff barrier. With FTA existing between two countries, one country opens its domestic market to goods originating from the other; FTA countries usually sacrifice benefits of some industries of own country in favor of its other industries. So that FTA is basically a compromise. Under FTA imports from preferential countries are usually free of tariff(or imposed very low tariff at the starting stage while finally would be expected zero tariff). To enjoy the preference the importing goods must be identified originating from FTA member countries, where the identification follows the rules of origin(ROOs) which are specified by detailed articles in FTA. By these specifications FTA member countries still aim to protect their domestic markets and manufacturers. The global economy has seen a surge in preferential trading agreements recently[6]. These include the North American Free Trade Agreement (NAFTA), the Central European Free Trade Agreement (CEFTA), the Australia-United States Free Trade Agreement (AUSFTA), and the Japan and Singapore New Age Economic Partnership Agreement (JSEPA). Many more continue to be shaped - the China-ASEAN Free Trade Agreement (CAFTA), for example.

Though details vary a lot among different FTA's and different goods, roughly speaking, ROOs can be divided into three categories: Value-Add(VA), Change in Tariff Classification(CTC) and Process rule. VA rule, also called local content rule, means value in the exported goods should contain value added in FTA member countries(usually exporting country) no less than a specified ratio. CTC rule means that if the product experiences a change in tariff classification from raw materials(perhaps semi-finished products) to final products in a country, then the product would be considered as originating from this country. Process rule is similar to CTC. It says that f a particular process, such as a particular chemical process, happens in a country then the product is viewed as made in this country. It is beyond our ability to discuss details of FTA in only one paper. To know details of a FTA we need definitely read the articles item by item, see NAFTA at *http://www.customs.gov/nafta* for example.

FTA is widely studied in macroeconomics from national, welfare and economic perspectives, such as [7], [10], [13], [3]. [8] made a detailed review on the relevant study in these areas. While on the influences of individual companies, little research has been done. [12] studies models where local content rules force the firm to purchase components from suppliers located in the country of manufacturer. [8] considers global sourcing under FTA where local content rules can either be satisfied or not which is determined by which option results in lower costs. Both the above two works consider a single-stage production.

In this work we consider a facility location problem under FTA. Specifically we focus on designing a production line for a multinational manufacturer under the existence of FTA. A multinational manufacturer has plants all over the world. Suppose its production line consists of many stages, each stage taking raw materials or the semi-products of previous stages as input, all of which may come from other countries. For example, to make a car, thousands of components are needed; many of them may come from different plants located among different countries. Please note that the plants in each stage may not belong to the manufacturer. In the North American (Canada, the US and Mexico) auto industry, for example, which operates under NAFTA, consider that a single vehicle can have in excess of 2000 parts which are supplied by hundreds of firms in a supply structure that spans North America and beyond. Automakers, including Asian transplants in Mexico, for example, who outsource production and/or import raw material and subassemblies are therefore faced with intricate facility location decisions which must factor tariff provisions under NAFTA. Different stages can be processed in various countries. However, different countries incur different production cost which may be an aggregation of all the possible costs, like labor cost,raw material cost, factory rental cost, in-land transportation, etc. If consecutive stages are conducted at different countries, necessary transportation will incur cost as well. A multinational company often needs to consider how to configure a production line on the globally located facilities for a new demand. Heuristic approaches, such as to locate the facilities to the closest places to the final market and to locate the facilities to utilize the cheapest resources, may not be optimal or even far away from optimum, due to the existence of various cost components. Most multinational companies face such a problems. Li&Fung is a successful Hong Kong based trading company. However, it works more like a supply chain designer. Facing a new demand from a customer, it will tailor the most cost-effective production line for this demand by utilizing all its raw materials and semi-products suppliers and manufacturers and by considering all kinds of governmental preferences. It has made a rapid development thanks to its flexible production line mechanism. See[9] for an introduction of this successful mechanism which is clearly becoming prevalent. This present paper is to study the production line design(PLD) problem under such settings.

The rest of the paper is structured as follows. Section 2 describes the problem formally in detail and gives notations used later on; a NP-hard proof for our problem is also given here. Section 3 presents an IP formulation for the problem and section 4 approaches the problem with a multi-exchange heuristic embedded in simulated annealing. Experimental results and analysis are given in section 5 and we finally summarize our work in section 6.

2 Problem Description

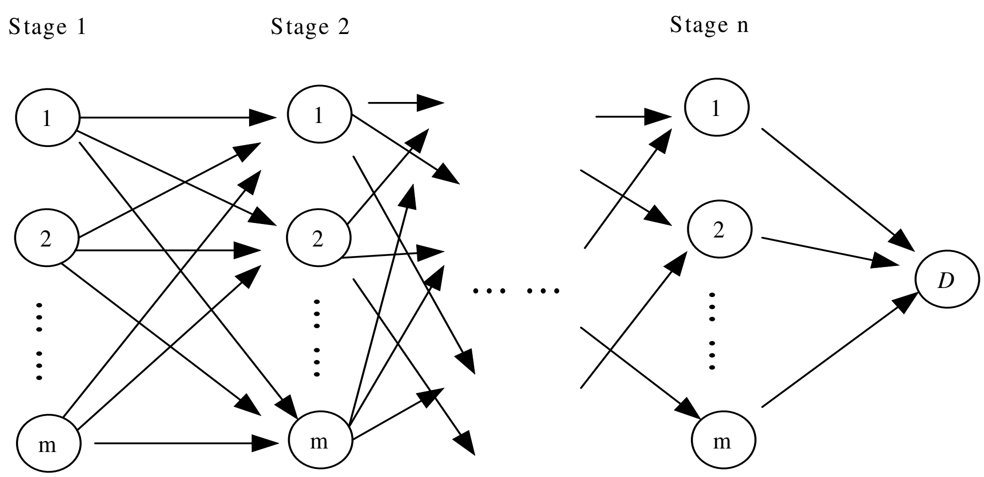

Figure 1: Illustration of problem setting

A multinational company opens a new market in country D for one product. Suppose the product is produced by a linear production line consisting of n stages. Please refer to Figure 1.

We currently only consider one single rule — Value-Add rule — for tariff elimination(In fact other rules can be viewed as special cases of VA rule as we will see later on). All the costs are defined on one unit of product(one unit is a generic concept, maybe one piece, one carton or one full truck load, etc). We assume the value of the product after stage k is V_k, $k = 1, ..., n$. Here V_k is a simplification of true value of a product, which may include all the costs incurred till now plus possible profit. However, factors like profit is hard to be modeled explicitly and moreover they are not important to the calculation so that we just project them in a value V_k which we call "internationally recognized value"(IRV). Suppose stage k occurs in country i and stage $k+1$ occurs in country j. Then the VA rule is as follows:

- if local contents ratio(LCR) in country i, $\frac{A_{ki}}{V_k} < \beta_{kij}$, tariff equal to $\alpha_{kij} V_k$ is imposed. Here A_{ki} is the sum of the added values in country i till stage k, either continuous stages or cumulative added values, which depends on particular FTA's; V_k is IRV of output from stage k. β_{kij} is the threshold LCR for tariff elimination if the production line goes to country j after stage k is finished in country i and α_{kij} is percentage of IRV paid as tariff if the VA rule is not satisfied.
- if locally added value in country i satisfies $\frac{A_{ki}}{V_k} \geq \beta_{kij}$, the product is free of tariff charge.

Moreover, the following cost parameters are given:

— m: number of countries.

— t_{kij}: unit transportation cost from country i to j, where $k = 1,..,n; i,j = 1,..,m,m+1$. We use $m+1$ to denote demand country.

— P_{kj}: production cost of stage k happening in country j, where $k = 1,..,n; j = 1,..,m$. Production cost here is a generic term, which may include various costs incurred in country k, such as raw material costs, labor costs, local production tax, facility cost, factory rental cost, etc.

The objective is to find an assignment of production stages to countries to minimize the total cost, including production cost, transportation cost and tariff.

2.1 Hardness of the Problem

In general there are two ways of computing local contents for VA rule, cumulative or not. When we calculate local contents of a country at one stage, under cumulative(also called outward processing) setting, we would consider the contents added in all the previous stages happened in this country; without cumulative rule, then we only consider the recent consecutive stages happened in this country. It's clear that outward processing regulation is more flexible and easier to satisfy.

PLP problem without cumulative rule can be solved by dynamic programming. Consider a stage k in country j, it's always preferred that the value(cost) of the incoming product is minimized: this benefits both total cost and higher possibility of satisfying local contents rule. Therefore just by considering all the countries stage by stage from left to right(in Figure 1), we can solve the problem by forward dynamic programming.

Theorem 1. PLD problem with cumulative rule applied in FTA is NP-hard.
Proof:

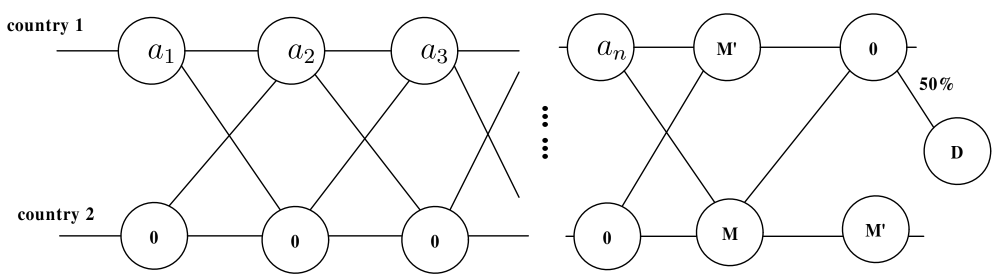

Figure 2: Proof of NP-hardness

The NP-hardness can be proved by reduction to the well-known NP-hard 2-partition problem: Given an integer set $\{a_1, a_2, ..., a_n\}$ with $\sum_{i=1}^{n} a_i = 2M$, can we find a subset S with $\sum_{a_i \in S} a_i = M$? We construct an instance of PLD problem as follows.

Suppose only two countries are available for choice. The production line consists of (n+2) stages. The transportation cost between these two countries are negligible; production costs are shown as Figure 2: From stage 1 to n in country 1 are a_1,a_2,...,a_n and in country 2 are 0, i.e., negligible; stage $(n+1)$ can only occur in country 2 with production cost M and stage $(n+2)$ can only occur in country 1 because we assume production cost $M' \gg M$. The threshold LCR is specified as follows: 0 for trading between country 1 and 2(that is no tariff between them); 50% for final export from country 1 to destination D. All these can be referred to Figure 2.

Suppose the tariff from country 1 to country D is pretty high without FTA. Therefore local contents rule for final product must be satisfied since final stage is carried out in country 1: the added value in country 1 is at least 50% of the total contents. We know stage (n+1) happens in country 2 with added value M, so the contents added in country 1 can not be less than M,i.e., $V_{n+2} = 2M$ is the possibly minimum cost. This requires that added value in country 1 from stage 1 to n is exactly M. After we solve the FTA problem, we will know immediately whether we can find a feasible solution to 2-partition problem: if the total cost is $2M$, then yes; otherwise no.

So PLD problem under FTA of cumulative rule is NP-hard. □

3 IP formulation

We first formulate the problem as an integer programming. Later on we provide a heuristic approach to solve the problem. Though IP formulation performs worse in terms of solution speed, it can provide optimal solutions or lower bounds(in cases of not reaching optimum due to insufficient computer memory even given enough time to run) for us to evaluate the performance of our heuristic method.

Decision Variables:

- I_{kj}: 1 if stage k occurs in country j; 0 otherwise. $k = 1, .., n;\ j = 1, .., m$
- J_{kij}: 1 if output of stage k is shipped from country i to country j. $k = 1, .., n;\ i, j = 1, .., m$.
- X_{kij}: tariff paid to country j if stage k occurs in country i and stage (k+1) occurs in country j, $i \neq j$; $X_{kii} = 0,\ k = 1, .., n;\ i, j = 1, ..., m + 1$

Objective Function:

$$\min \sum_{k=1}^{n}\sum_{j=1}^{m} I_{kj}P_{kj} + \sum_{k=1}^{n-1}\sum_{i=1}^{m}\sum_{j=1}^{m} J_{kij}t_{kij} + \sum_{k=1}^{n-1}\sum_{i=1}^{m}\sum_{j=1}^{m} X_{kij} + \sum_{i=1}^{m}(t_{ni,m+1}I_{ni} + X_{ni,m+1}) \qquad (1)$$

Constraints:

1. Each stage must be assigned to exactly one country.

$$\sum_{j=1}^{m} I_{kj} = 1, \qquad k = 1, .., n \qquad (2)$$

2. Make sure I_{ki} and J_{kij} are consistent.

$$I_{ki} = I_{k+1,j} = 1 <=> J_{kij} = 1$$

<=>

$$J_{kij} + 1 \geq I_{ki} + I_{k+1,j}, \quad k = 1, .., n-1; i, j = 1, .., m, i \neq j \qquad (3)$$

3. Tariff calculation.

$$\frac{\sum_{\kappa=1}^{k} I_{\kappa i}P_{\kappa i}}{V_k} < \beta_{ij} \ \&\ J_{kij} = 1 => X_{kij} = \alpha_{ij}V_k$$

<=>

$$h_{kij}M \geq \beta_{ij}V_k - \sum_{\kappa=1}^{k} I_{\kappa i}P_{\kappa i} \text{ where } h_{kij} \in \{0, 1\} \qquad (4)$$

$$X_{kij} - \alpha_{kij}V_k \geq G(h_{kij} + J_{kij} - 2) \text{ where } G \text{ is a large number}$$

$$k = 1, .., n, i, j = 1, .., m; \text{ when } k = n, i = 1, .., m; j = m + 1 \qquad (5)$$

Note:

As specified previously, VA rule can be viewed as the most widely used rule because other rules like "CTC" and process rules can be viewed as special cases of VA rule. Suppose FTA specifies that some stages with "CTC" or involving specific chemical processes: If a stage k occurs in country i then the product can be considered as originating from this country i. This is equivalent to the setting of local content critical ratio $\alpha_{kij} = 0$, or directly $X_{kij} = 0$ in above (1) – (5).

4 A Multi-Exchange Heuristic Embedded in Simulated Annealing

In this section we approach the PLD problem with a multi-exchange heuristic embedded in simulated annealing. Our multi-exchange neighborhood local search is a variant of the very large-scale neighborhood(VLSN) search. VLSN is an endeavor to explore a neighborhood as large as possible while keeping the neighborhood search procedure fast. See [1] for a good survey on VLSN. Our method(named "SAVLSN") contributes to the literature in two aspects: (1) we search the neighborhood heuristically based on an constructed estimation improvement graph ([2] constructs exact improvement graphs for the neighborhoods) and (2) we embed VLSN into simulated annealing; to the best of our knowledge, this is the first work on this combination. Simulated Annealing(SA) is a meta-heuristic that differs from the traditional hill-climbing search in the sense that it may accept a down-hill move which can decrease the quality of the objective function with a certain probability related to the temperature variable[4]. The algorithm is outlined in Algorithm 1. Here we use the geometric annealing scheme in the framework with the constant C_0 to be 0.995. One thing to note here is we also use a reheating mechanism whenever an iteration could not yield a new current solution. The reheating mechanism is to counter the effect of annealing in the hope to have a higher chance of accepting diversing local moves in later iterations. The reheating is also geometric defined to be $Temperature = Temperature * (1 + (1 - C_0)/5)$. From our test we found that once reheating is used, the result quality can be improved by 1%-1.5% on average.

Algorithm 1 SAVLSN Outline

```
read input: n, m, V_k, t_kij, P_kj, α_kij, β_kij
S ← Weighted Probablistic Initial Solution Generation
Temperature ← T_max; Iter ← 0
while Iter < Max_Iter and
Temperature > T_Terminate do
  with probability 0.5
   Stemp ← VLSN_Cycle(S, random(2, K_max))
  with probability 0.5
   Stemp ← VLSN_Path(S, random(2, K_max))
  δ = value(Stemp) − value(S)
  if δ ≤ 0 then
    S ← Stemp
  else
    p = e^(−δ/Temperature)
    with probability p
     S ← Stemp
    with probability 1 − p
    reheat()
  end if
  if value(S) > best_value then
    best_value ← value(S);
  end if
  iter ← iter + 1
  Temperature ← Temperature * C_0
end while
```

4.1 Initial Solution generation

Let the array S of length n represent a solution where $S[i]$ is the index of country which stage i is assigned to, $1 \le S[i] \le m, 1 \le i \le n$. We used two methods to generate the initial solution. The first one is to randomly choose a country for a stage to be processed in, serving as a comparison methodology for the second method. The second one is a weighted probability method: in order to assign a country index to every stage, we would consider the stages from stage 1 to stage n sequentially. Since there is no tariff cost or transportation cost involved in stage 1 of the production by the FTA context, we estimate the total cost of stage 1 if assigned to country j, $1 \le j \le m$ to be production cost P_{1j}. The estimation is based on the ignorance of the influence of assigning country index to stage 1 on later decisions for stage 2 to stage n. Define $Q_{1j} = 1/P_{1j}$ and $Q_{total} = \Sigma_{j=1}^{m} Q_{1j}$. Then our weighted probability generator would assign

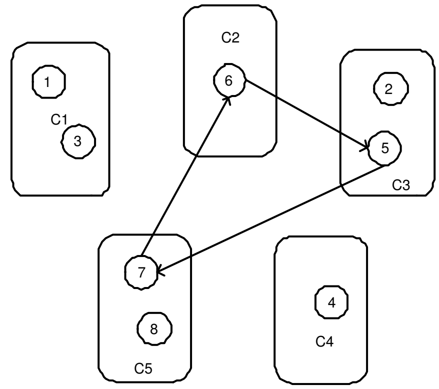

Figure 3: VLSN cyclic exchange example with $n = 8,\ m = 5,\ K = 3$

index $j, 1 \leq j \leq m$ to stage 1 with probability Q_{1j}/Q_{total}. This is to encourage stage 1 to be processed in countries that have a smaller production cost by allowing that particular country to be selected with a larger probability. After stage 1 is assigned to a country, we will continue to decide the country indexes for stage 2 to n in a similar fashion sequentially except that the estimated cost for assigning country index j to stage i would in addition includes the transportation cost and tariff (if not exempted) from country where stage $(i-1)$ is processed to country j. To decide country index for the last stage, the tariff and transportation cost for the last stage to the destination is also taken into consideration.

4.2 Very Large-Scale Neighborhood Search

Given a solution $\mathbf{S}$, the neighborhood $N(\mathbf{S})$ is defined as the set of all feasible solutions $\mathbf{S}'$ that is achievable from $\mathbf{S}$ by a single neighborhood move. In general, the larger the neighborhood size $|N(\mathbf{S})|$ is, the better the solution quality will be after one local move. However, it is also often the case that due to a very large number of neighborhood solutions, the running time to perform one neighborhood move is costly and can result in overall inefficiency. The idea of VLSN stems from maintaining a large set of neighborhood solutions while explore them efficiently. In this paper we will define the cyclic and path neighborhood exchange moves as the local move methods.

4.2.1 Neighborhood Structure

For a solution $\mathbf{S}$, define set C_j $1 \leq j \leq m$ as $C_j = \{i|S[i] = j, 1 \leq i \leq n\}$, which is the set of indexes of stages processed in country j. A cyclic exchange neighborhood move is to first select K different countries $i_1, i_2, ..., i_K$ such that $C_{i_j} \neq \emptyset, \forall j \in \{1, 2, .., K\}$. In each selected country j $1 \leq j \leq K$, choose stage $t_j \in C_{i_j}$. And then reassign stages t_1 to t_K to country $C_{i_j}, j = 1..K$ in a cyclic manner: $S[t_i] := S[t_{i+1}]$ for $i = 1..K-1$, and $S[t_K] := S[t_1]$. Consequently, C_j, $1 \leq j \leq m$ is changed accordingly. The changes take place simultaneously. For the simple example demonstrated in Figure 3, after the local move three set C_2, C_3 and C_5 are changed with the new C_2={7}, C_3={2, 6}, C_5={5, 8} while C_1 and C_4 remain unchanged. It is obvious that by the K-cyclic change, the number of neighborhood solution is $(n/K)^K K!$ assuming the n stages are uniformly allocated in m countries and in general $N(\mathbf{S}) = \Omega(n^K)$. When we allow K to be linear with n the neighborhood size becomes exponential to n. This is also why such kind of neighborhood moves are called very large-scale neighborhood search. In our algorithm we fix K_{max} to be approximately 10% as large as m, and in each iteration of cyclic local move we would randomly select K in the range [2, K_{max}]. The path exchange neighborhood is very similar to the cyclic one in spite that the path exchange would not select any stage in C_{i_K} to move it to C_{i_1}.

In order to choose K proper stages involved in a cyclic exchange local move, we need to have a way to specify our estimated total cost change if we choose some stages in the local move. The idea of an improvement graph originated in [2] is a practically nice and easy-to-implement way to reflect such estimations. We would develop our own "estimation" improvement graph as it differs from in [2] where the arc weights really reflect the exact cost change of stages. Cost calculation of the PLD problem is impossible with partial information. The arc weights in our improvement graph are just estimations. We will discuss how to construct the estimation improvement graph in the next section.

4.2.2 Estimation Improvement Graph

Given a solution **S** and C_j, $1 \le j \le m$ defined in the previous sections, an *estimation improvement graph* is a directed graph $G(\mathbf{S}) = (V, E)$. The set of vertices V contains n nodes: v_k, $1 \le k \le n$ each representing a stage k in the solution **S**. The arc set E represents the relation among different stages. There is a directed arc (k, l) from v_k to v_k if and only if $S[k] \neq S[l]$. We define the weight on arc (k, l) to be E_{kl} where:

$$E_{kl} = \begin{cases} P_{k,s[l]} - P_{l,s[l]} & , \texttt{ if } k = 1 \texttt{ or } l = 1 \\ P_{k,s[l]} - P_{l,s[l]} + t_{k,s[k-1],s[l]} - t_{l,s[l-1],s[l]} + X_{k,S[k-1],[S[l]} - X_{l,S[l-1],S[l]} & , \ o.w. \end{cases}$$

The above arc weight is designed to reflect the total cost change if we reassign stage k to country $S[l]$ and reassign stage l to somewhere else. The above function E_{kl} can only be an estimation because the exact transportation cost and tariff cost for stage k in country $S[j]$ are not available until all stages $1, .., k-1$ are fixed. But these $k-1$ stages may involve in the current cyclic/path exchange and therefore can not be fixed yet.

4.2.3 Probabilistic Selection of Cycles and Paths

Once we have defined the estimation improvement graph, our algorithm will first randomly choose K countries C_{i_j}, $1 \le j \le K$. If it is a cyclic neighborhood move we must enforce $C_{i_j} \neq \emptyset$, $\forall j \in \{1, 2, .., K\}$, while if it is a path exchange we would allow $C_{i_K} = \emptyset$. In the neighborhood search we will first choose a stage in C_{i_1} to include as the first stage in the cycle/path as follows: Let the production cost P_{ji_1} be the indicator of our preference to choose stage j originally in C_{i_1} for all j such that $j \in C_{i_1}$. Define $P_{total} = \Sigma P_{ji_1}$. Then stage $j \in C_{i_1}$ would be selected by the cyclic neighborhood change with probability P_{ji_1}/P_{total}. This procedure is similar to when we generate the country index for the first stage in the weighted probability initial solution generation section. However, there are two differences. The first difference is that because here we are selecting stages to "move away" from the currently assigned countries, we would assign a larger probability to stages that have a large production cost rather than those have a small production cost as in initial solution generation. The second difference is that here we are calculating the probabilities to select one stage when the country index C_{i_1} is fixed, not the other way round as in the initial solution generation when the stage is fixed and we search for the country index.

When stage from country C_{i_1} is selected, we will sequentially select one stage for each of the remaining $K-1$ countries to participate in the cyclic exchange move. Let the index of the selected stage from country C_{i_j} be l_j for $1 \le j \le K$. The selection of l_j is based on the value of l_{j-1} for $2 \le j \le K$. In the estimation improvement graph, there should be an arc from the node representing l_{j-1} to every node in C_{i_j} by definition. A negative arc weight indicates a potential improvement in solution quality if we make the exchange local move to contain the stages associated by this arc. We modify the arc weights in the following way: first multiply them by -1 and then add a minimum positive number to all those arcs to make the arc weights all positive. For example, originally we have weights $\{1, 2, 3, -4, -5\}$. Then the procedure will be first change them to $\{-1, -2, -3, 4, 5\}$ and then to $\{3, 2, 1, 8, 9\}$. This is to facilitate later calculations of probabilities for each stage in C_{i_j} to be selected. Let E'_{pq} be the modified arc weight from stage p to stage q and $Arc_{totoal} = \Sigma_{q \in C_{i_j}} E'_{l_{i-1}q}$. We would select stage $q \in C_{i_j}$ in the cyclic exchange neighborhood move with probability $E'_{l_{i-1}q}/Arc_{totoal}$. As we need to note here, if it is a cyclic exchange, when deciding stage from C_{i_K} we would also consider the cost from C_{i_K} to C_{i_1} by including the arc weight E_{ql_1} to calculate the probability to select the stage $q \in C_{i_K}$. If the local move is a path exchange, we do not need to select stage from C_{i_K} as it is unnecessary and C_{i_K} may also be empty. When l_i $\forall i \in \{1, 2, ..K\}$ are fixed, we would perform the cyclic/path exchange to complete an iteration of the neighborhood search as described in section **4.2.1**.

5 Experimental Analysis

5.1 Test Instances Generation

As the PLD problem is very new, to the best of our knowledge there has not been any previous research done to address the PLD problem which takes FTA into consideration in the literature and hence no benchmark test sets established. In order to experiment on our method, we have developed a test instance generator which accommodated some real-world concerns which we believe realistic. These include:

1. The product value is generally increasing as more stages have been processed. For $i \in$\{1, 2, 3\}, we specify the following parameters: S_i, the base standard product value for the ith one third stages; inc_i,

n	m	$\mu_{SA_{0.1}}$	$\sigma_{0.1}$	t_1	$\mu_{SA_{0.5}}$	$\sigma_{0.5}$	t_2	δ
60	30	8.02	0.34	1.21	8.13	0.35	3.16	1.37%
80	30	114.04	5.86	1.25	117.55	7.85	4.29	3.04%
100	50	1740.83	48.61	2.25	1840.11	78.05	5.85	5.70%
100	80	1765.77	82.17	2.49	1924.26	90.09	22.34	8.98%
120	50	2489.85	698.92	14.39	27446.21	1147.35	35.11	10.23%

Table 1: Comparison between $SAVLSN_{0.1}$ and $SAVLSN_{0.5}$

the exponential increment on product value compared with the previous stage; d_i and u_i, the amount of random fluctuation in product value. For stages 1 to $\lfloor n/3 \rfloor$ which belong to the first one third stages, the IRV of output of stage j is defined as

$$V_j = S_1 * (1 + inc_1)^{(i-1)} * \texttt{Unif}[1 - d_1, 1 + u_1], j \in \{1, 2, ..\lfloor n/3 \rfloor\}$$

Unif[x,y] generates a real number uniformly in the region [x,y]. IRV for the remaining stages are calculated in the same way with respective parameters.

2.Some countries may have a relative low production cost(such as labor intensive work in Asia) for early stages, while other countries may benefit later stages. We model this situation by introducing the fluctuation range parameters d_{ij} and u_{ij} where i ∈ {1..n} and j ∈{1..m}. d_{ij} and u_{ij} specifies how much fluctuation the production cost of stage i in country j can have compared with V_j, as we define the production cost

$$P_{ij} = V_i * \texttt{Unif}[1 - d_{ij}, 1 + u_{ij}], \forall i \in \{1..n\},\ j \in \{1..m\}$$

The n*m matrixes **d** and **u** are assigned values such that we can model the preference of countries for different stages. For example, if country j is in favor of first half of the stages of process, we would assign a higher d_{ij} and lower u_{ij} value for stages 1..$\lfloor n/2 \rfloor$ in country i to model this situation.

3. Each country location is modeled as a pair of (x,y) coordinate in a bounded square randomly in a 2D plane. The transportation cost t_{kij} is proportional to the product of the distance between countries i and j, and product value V_i, with at most 10% fluctuation.

4. Tariff ratio and threshold LCR are specified by the given parameters tar and lcr which ranges from 20% to 50%. Both tariff ratio and threshold LCR are subject to 40% fluctuation.

5.2 Experiment Analysis on Parameter K_{max}

In order to evaluate the performance of the SAVLSN algorithm, we will investigate the effect of the most important parameter of the VLSN local search method, i.e, K_{max} on the overall performance. We compare the performance of SAVLSN with K_{max}=0.1n and K_{max}=0.5n. Since there are no previously established benchmark set, we generated 5 groups of test sets with varying scales. Each test set group consists of 20 test instances genertaed according the the categories in the previous section. For the sake of having a feel of the effectiveness of the VLSN used in the simulated annealing framework, we will also compare the acceptance rate of the local move in the framework for different iteration ranges. All the experiments are done on a machine with PIV 1.4GHZ CPU and 256M memory.

The experimental results are listed in Table 1, where $\mu_{SA_{0.1}}$ and $\mu_{SA_{0.5}}$ are the average costs obtained by the SAVLSN algorithm using 0.1n and 0.5n as the K_{max} value respectively; σ_{SA} measures the standard deviation of the results for the 20 test instances in each group, and t_1 and t_2 are the average running time in second for each instance; δ measures the percentage of result quality of $SAVLSN_{0.1}$ over $SAVLSN_{0.5}$. From the table, we clealy see that setting K_{max}=0.1n is more competitive as it provides 1%-10% better results compared with K_{max}=0.5n. This is due to the reason that a too large estimated cycle length would easily lose the advantage of small adjustment with K=2 for example (which is actually the widely used 2-opt operation in combinatorial optimization problems). In addition, the standard deviation of results is also higher than that for $SAVLSN_{0.1}$. Noticeably enough, both parameter settings result in efficient running time for less than 40 seconds while $SAVLSN_{0.1}$ is even more time-efficient to obtain the results within 15 seconds. The longer running time for $SAVLSN_{0.5}$ is within expectation as the longer cyclic local search each operation needs to perform will naturally result in longer running time. In general, according to our experiments for different PLD problem variations, setting K_{max} to roughly equal $0.1n$ is practically good in the VLSN context.

To further analyze the use of simulated annealing as a framework for $SAVLSN_{0.1}$ and $SAVLSN_{0.5}$, we recorded the acceptance rate of VLSN local move in simulated annealing framework for iterations 1 to 1000 without reheating because it can affect the natural acceptance rate of the algorithm we want to investigate. We record the acceptance rate for every 100 iterations performed. The results for the 20 test instances for n=100 and m=50 are found in Figure 4. Each point with x-axis value i represents the acceptance rate for iteration 1 to $100i$. Other groups of test instances have similar acceptance rates. According to [11], the initial acceptance rate of SA should be around 60%. This again explains why $SAVLSN_{0.1}$ performs better than $SAVLSN_{0.5}$ since $SAVLSN_{0.1}$ has an initial acceptance rate near to 65% while its counterpart is with that of about 28%.

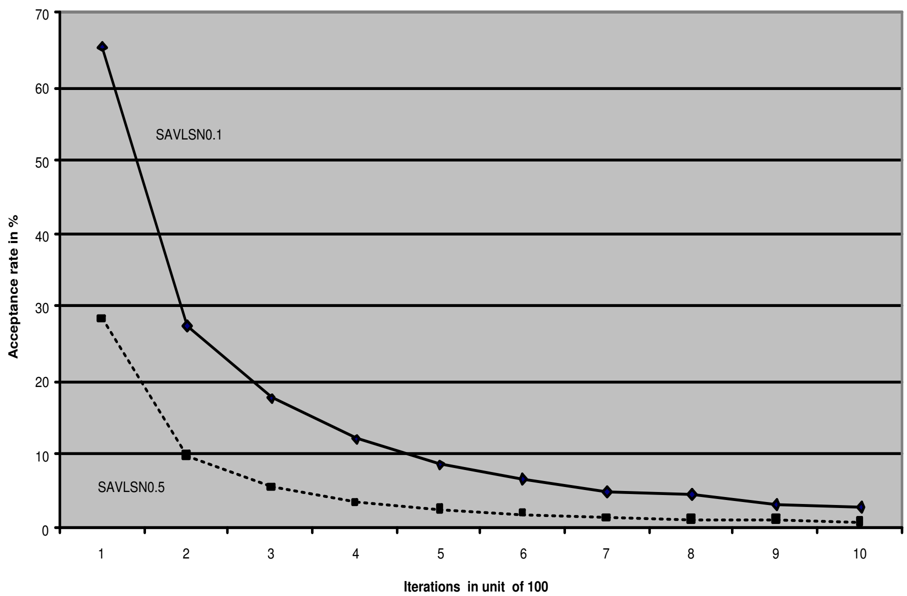

Figure 4: Acceptance Rate for first 1000 iterations

5.2.1 Initial Solution Generation Mechanism Comparison

We used a weighted probability method to generate the initial solution. We would compare it with a random generation method. We use an instance with $n = 80$ and $m = 30$ to compare the algorithm performance. All parameters are assigned the same values except the initial solution generation mechanism of the two runs. Both have similar time efficiency, with a run time of 17.38 and 19.23 seconds for the random and weighted probability methods respectively. The final result of the weighted probability method is 11.3% better than the random method. However, this gap is not important as in this comparison we only let each method run a relatively small amount of time, and we can often improve the result quality by multi-restart as each restart would only require a few seconds. Here we do not allow multiple restarts because we want to focus on the comparison of solution convergence efficiency and hence we need to fix one initial solution in each run. The experimental results are presented in Figure 5.

The x-axis is the running time used by each method at that point, and the y-axis is the difference of current solution to the best solution the weighted probability method obtained in percentage. The result at point with $x = 0$ is the initial solution generated. It is not surprising that weighted probability would result in a better starting solution, as it takes into the various costs into consideration. Actually both converge very fast$--$they converge to 10% the final result within 3 seconds. Weighted probability provides better performance on both result convergence rate and result quality.

5.3 Experiment Analysis on CPLEX and SAVLSN

ILOG CPLEX is one of the most widely used Integer Programming solver. We make a comparison between our SAVLSN and CPLEX 9.0. In this section the K_{max} for SAVLSN is $0.1n$. From our experiments we find that even both m and n are of moderate size (40 to 60), CPLEX could not obtain a feasible solution even given long enough time due to the large scale of the IP model. Therefore firstly we limit the test instance in the largest possible range that CPLEX could reach a feasible solution. We have generated 2 groups of small

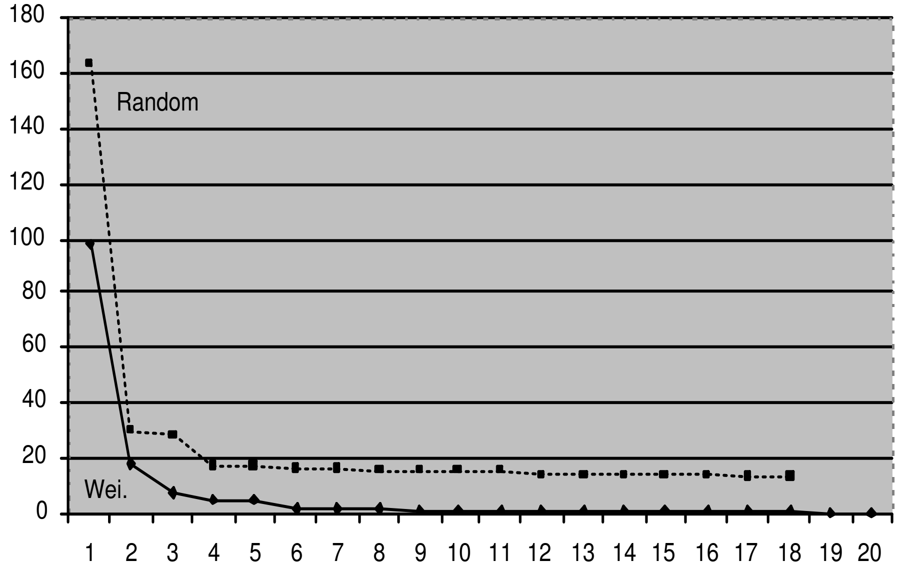

Figure 5: Result convergence for 2 initial solution generation methods

test instances, each with 21 test cases starting from n=5, m=5 and increase step by step until n=30 and m=30. We set a time limit 10,000 seconds for CPLEX running time. If after 10,000 seconds CPLEX could not generate optimal solution, the best solution obtained at that moment is reported. Instances of the first group have fixed tariff ratio 30% and threshold LCR 30% , both with up to 20% random variation, while instances of the second group have tariff ratio 50% and threshold LCR 50%, also with up to 20% random variation. The results of the first 21 instances are in Table 2, where t_1 and t_2 are the time used by CPLEX and SAVLSN respectively. If CPLEX cannot finish within 10,000 seconds the t_1 column is indicated by a hyphen and the lower bound obtained by CPLEX at 10,000 seconds is presented in the "LB" column. Ratio_1 is the ratio of cost by SAVLSN over that by CPLEX. If CPLEX optimal result is generated within 10,000 seconds, Ratio_2 is indicated by a hyphen, otherwise it is the ratio of cost by SAVLSN over that of the lower bound.

From this table we can see that SAVLSN is a competitive heuristic method on the PLD problem. With respect to solution quality, No solutions given by SAVLSN are 5% worse than solutions obtained by CPLEX. Among them, 9 instances (42.9%) are optimal; 3 instances (14.3%) are better than CPLEX's solutions in 10,000 seconds by a percentage from 0.3% to up to 7%; 6 instances (28.6%) are within 1% CPLEX's result, and only 3 instances are worse than CPLEX's results by 1% to 5%. Nevertheless, CPLEX obtained 15 optimal results out of the 21 instances. For instances where optimal results are not obtained, SAVLSN are always within 10% difference from the lower bound, which is an absolute guarantee of result quality. Taking algorithm efficiency into consideration, SAVLSN performs much more stable by always providing the result within 10 to 25 seconds, while CPLEX's time requirement is roughly exponential to the size of instance. For the 9 instances where CPLEX get better results than SAVLSN by 0.11% to 4.08%, CPLEX is using 8 to 545 times the run-time required by SAVLSN. In addition, considering the heuristic nature of SAVLSN and that we limit the instance size so that CPLEX could start to optimize, it is fair to say that SAVLSN is a competitive heuristic when compared with CPLEX IP Solver on the FTA problem.

The results of the second group experiments are presented in Table 3 in a similar fashion. For this group of test instances, SAVLSN performs slightly worse than the previous one, as for 2 instances (25_15, 30_20) SAVLSN's result is more than 5% worse than that of CPLEX. This, could be the consequence of the fact that high tariff and LCR ratios tend to encourage many stages to process in the same country to avoid transportation cost and tariff, so in the good solutions the number of different countries involved in the production is small compared with m. However, SAVLSN is using cyclic/path exchange local move. Although the path exchange local move could possibly decrease the number of cities involved for processing the n stages, cyclic exchange move never decreases the number of countries involved. SAVLSN has this disadvantage and hence it is not as efficient as CPLEX in exploring the regions where only a very small

Size	CPLEX	t_1	LB	SAVLSN	t_2	Ratio$_1$	Ratio$_2$
5_5	224.772	1.00	-	224.772	10.98	100.00%	-
10_5	440.257	1.00	-	440.257	26.21	100.00%	-
10_10	455.606	6.00	-	455.606	19.93	100.00%	-
15_5	692.283	1.00	-	693.864	15.15	100.23%	-
15_10	711.747	6.00	-	711.747	14.89	100.00%	-
15_15	580.890	17.00	-	580.890	16.22	100.00%	-
20_5	1481.703	1.00	-	1481.703	17.89	100.00%	-
20_10	1527.546	105.00	-	1527.546	16.55	100.00%	-
20_15	1394.503	461.00	-	1394.503	17.48	100.00%	-
20_20	1078.740	1353.00	-	1122.738	17.48	104.08%	-
25_5	2000.440	1.00	-	2003.950	21.74	100.18%	-
25_10	2203.780	157.00	-	2206.119	20.35	100.11%	-
25_15	1768.260	1345.00	-	1822.545	19.96	103.07%	-
25_20	1686.560	-	1683.990	1726.465	18.36	102.37%	102.52%
25_25	1800.390	-	1699.850	1794.241	19.12	99.66%	105.55%
30_5	2577.530	5.00	-	2577.534	23.40	100.00%	-
30_10	2620.060	599.00	-	2631.067	23.52	100.42%	-
30_15	2229.780	6936.00	-	2247.729	22.30	100.80%	-
30_20	2474.420	-	2301.620	2477.636	21.34	100.13%	107.65%
30_25	2652.290	-	2394.690	2600.158	21.61	98.03%	108.58%
30_30	2423.700	-	2117.080	2258.655	22.60	93.19%	106.69%

Table 2: Experimental results on instances with 30% tariff and LCR ratio

Size	CPLEX	t_1	LB	SAVLSN	t_2	Ratio$_1$	Ratio$_2$
5_5	293.538	1.00	-	293.538	37.99	100.00%	-
10_5	402.631	1.00	-	402.631	24.11	100.00%	-
10_10	424.003	3.00	-	424.003	24.48	100.00%	-
15_5	770.252	1.00	-	770.252	30.87	100.00%	-
15_10	749.333	63.00	-	749.333	55.62	100.00%	-
15_15	629.977	47.00	-	629.977	28.77	100.00%	-
20_5	1291.777	2.00	-	1291.777	35.06	100.00%	-
20_10	1282.450	14.00	-	1295.130	34.32	100.99%	-
20_15	1323.949	299.00	-	1323.949	34.16	100.00%	-
20_20	1287.680	1947.00	-	1308.052	33.51	101.58%	-
25_5	2149.300	4.00	-	2200.713	39.16	102.39%	-
25_10	1957.030	407.00	-	1963.642	38.31	100.34%	-
25_15	1886.950	744.00	-	2041.687	37.58	108.20%	-
25_20	1850.320	4807.00	-	1890.099	45.74	102.15%	-
25_25	1729.000	-	1698.430	1800.747	43.82	104.15%	106.02%
30_5	2542.350	14.00	-	2547.072	42.63	100.19%	-
30_10	2856.920	1163.00	-	2871.363	44.22	100.51%	-
30_15	2430.150	-	2353.270	2440.994	44.62	100.45%	103.73%
30_20	2288.420	-	2144.240	2452.440	42.01	107.17%	114.37%
30_25	2465.600	-	2218.340	2518.320	41.62	102.14%	113.52%
30_30	2618.950	-	2346.150	2573.357	45.27	98.26%	109.68%
40_10	6563.950	3573.00	-	6772.009	27.81	103.17%	-
40_15	7406.240	-	7168.940	7414.705	26.16	100.11%	103.43%

Table 3: Experimental results on instances with 50% tariff and LCR ratio

number of countries is involved compared with the number of stages.

In addition, as we mentioned at the beginning of this section, CPLEX failed to generate satisfactory results for instances with slightly larger size. We used 3 additional instances 40_40, 50_50 and 60_60 with tariff and LCR ratio fixed at 30%. CPLEX runs 10,000 seconds for both the first two insances and could not find optimal solution. For the 60_60 test instance CPLEX even could not reach any feasible solution while SAVLSN can obtain its result in . For the 40_40 instance CPLEX's result is 4.6% worse than SAVLSN result which was obtained within 40 seconds, and for the 50_50 instance CPLEX's result is 22.8% worse than SAVLSN's result in 48 seconds. More over, CPLEX does not converge fast as there are 27% and 32% gap between CPLEX's result and the lower bound CPLEX calculated for the 40_40 and 50_50 instances respectively. On the other hand SAVLSN can handle larger test instances much more satisfactorily than CPLEX. We conclude that for real applications where hundreds of stages and tens of countries are involved, SAVLSN is a suitable approach to PLD problem.

6 Summary

In this paper, we solved a production line design problem under FTA with a multi-exchange heuristic. The problem can be viewed as a facility location problem while FTA brings new cost factors into it and various rules of origin such as VA rule have much complicated the problem. We proposed a heuristic method called SAVLSN to tackle the NP-hard problem. The method embedding a very large-scale neighborhood search in a simulated annealing framework showed quite stable and good performance by comparing with solutions of an integer programming model solved by CPLEX.

References

[1] R.K. Ahuja, O. Ergun, J.B. Orlin, and A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematic*, 123:75–102, 2002.

[2] R.K. Ahuja, J.B. Orlin, S. Pallottino, M.P. Scaparra, and M.G. Scutella. A multi-exchange heuristic for the single source capacitated facility location. *Management Science*, 50(6):749–760, June 2004.

[3] J Bair and G Gereffi. Upgrading, uneven development, and jobs in the north american apparel industry. *Global Networks - A Journal of Transnational Affairs*, 3(2):143–169, Apr 2003.

[4] K. Dowsland. *Simulated annealing*, pages 20–69. Blackwell Scientific Publications, Oxford, 1993.

[5] Richard L Francis, Leon F. McGinnis, and John A. White. *Facility Layout and Location: An Analytical Approach*. Prentice Hall, second edition, 1993.

[6] J. Ju and K. Krishna. Firm behaviours and market access in a free trade area with rules of origin. *Working paper 6857*, 1998.

[7] Anne O. Krueger. Free trade agreements as protectionist devices: Rules of origin. NBER working paper, April 2003.

[8] Yanzhi Li, Andrew Lim, and Brian Rodrigues. Global sourcing under free trade regimes. *submitted for publication*, 2004.

[9] Joan Magretta. Fast, global, and entrepreneurial: Supply chain management, Hong Kong style: An interview with Victor Fung. *Harvard Business Review*, Oct. 2002.

[10] KM Mann. Recent developments in government operations and liability - united mexican states v. metalclad corporation: The north american free trade agreement provides powerful private right of action to foreign investors. *Urban Lawyer*, 35(4):697–702, Fal 2003.

[11] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000.

[12] Charles L. Munson and Meir J. Rosenblatt. The impact of local content rules on global sourcing decisions. *Production and Operations Management*, 6(3):277–290, 1997.

[13] S Stordal. Impacts of the european economic area agreement on the structure and concentration of roundwood sales in norway. *Forest policy and economics*, 6(1):49–62, Jan 2004.