Principal Investigators: Leyuan Shi and Robert R. Meyer
Institution: University of Wisconsin–Madison
Award Number: DMI-0100220 Program: DMII-OR
Project Title: GOALI:Global Meta-Hybrids for Supply Chain Optimization

Large-Scale Supply Chain Optimization via Nested Partitions

Leyuan Shi
leyuan@ie.engr.wisc.edu
Industrial Engineering Department
University of Wisconsin–Madison

Robert R. Meyer
rrm@cs.wisc.edu
Computer Sciences Department
University of Wisconsin–Madison

Mehmet Bozbay
bozbay@cae.wisc.edu
Industrial Engineering Department
University of Wisconsin–Madison

Winston C. Yang
winston@cs.wisc.edu
Computer Sciences Department
University of Wisconsin–Madison

October 15, 2002

Abstract

Large-scale supply chain design problems are generally intractable when dealt with as single entities via general purpose branch-and-bound solvers such as CPLEX. In this research we demonstrate that the nested partition (NP) method is capable of efficiently producing very high quality solutions to large-scale supply chain problems by taking advantage of their structure to generate an appropriate set of restricted subproblems. The implementation of NP that we describe here first employs an optimization-based heuristic to generate an efficiency ranking of the available warehouses and then employs biased sampling (from the ranked warehouse set) guided by the NP global optimization framework. The overall process is implemented using the looping

constructs available in the AMPL modeling language and employs the CPLEX mixed-integer branch-and-cut code to solve the optimization subproblems that arise. This implementation represents a novel use of CPLEX as an evaluator of samples generated using a global view that takes advantage of the structure of the original problem. Our computational results demonstrate that NP also outperforms specialized approaches such as the Lagrange relaxation methods that previously have been regarded as the most effective techniques for large-scale supply chain optimization.

1 Introduction

We consider large-scale supply chain problems with the following properties:

- A set of plants and customers are geographically dispersed in a region.
- Each customer experiences a demand for a variety of products that are manufactured at the plants. Products are shipped from plants to warehouses and then distributed to customers. For each shipping link and product there is a per unit shipping cost.
- A given number of warehouses must be located in the distribution network from a list of potential sites. A fixed cost must be paid for each warehouse that is opened.

The goal is to *identify the W best warehouse locations (for any specified value of W)* that allow satisfaction of each customer's *demand* for each product from a single warehouse, *minimizing total cost*. This particular format for supply chain optimization problems was motivated by data for a very large real world application furnished to us by Rockwell Automation (Milwaukee, Wisconsin). The actual test problems considered below were generated via suitable approximations to the data distributions provided by Rockwell.

Large-scale problems of this type are intractable when they are input as single problems to general purpose branch-and-bound solvers such as CPLEX. In this research we demonstrate that the nested partition (NP) method ((Shi, et al., 1999) and (Shi and Olafsson, 2000)) is capable of efficiently producing very high quality solutions to large-scale supply chain problems by taking advantage of their structure to generate an appropriate set of restricted subproblems. The implementation of NP that we describe here first employs an optimization-based heuristic to generate an efficiency ranking of the available warehouses and then employs biased sampling (from the ranked warehouse set) guided by the NP global optimization framework. The overall process is implemented using the looping constructs available in the AMPL modeling language and employs the CPLEX mixed-integer programming (MIP) solver for the optimization subproblems that arise. This implementation represents a novel use of CPLEX as an evaluator of samples generated via a global view that takes advantage of the structure of the

original problem. Our computational results demonstrate that NP also outperforms specialized approaches such as the Lagrange relaxation methods that previously have been regarded as the most effective techniques for large-scale supply chain optimization.

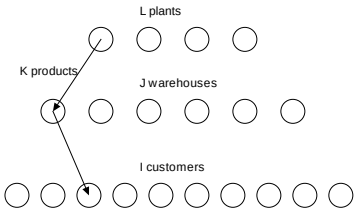

Figure 1: Products go from plants to warehouses to customers.

2 The supply chain model

We formulate the problem as a mixed integer program. The indices and sets are as follows:

- i (I) customer(s)
- j (J) warehouse(s)
- k (K) product(s)
- ℓ (L) plant(s)
- W warehouses to locate

The problem data are as follows:

parameter	description
$c_{\ell jk}$	unit shipping cost from plant ℓ to warehouse j of product k
d_{jik}	unit shipping cost from warehouse j to customer i of product k
f_j	fixed cost of opening and operating a warehouse at site j
w_{ik}	demand of customer i for product k
s_k	volume of unit of product k
q_j	capacity (in volume) of warehouse located at site j
$v_{\ell k}$	supply at plant ℓ of product k

The problem variables are as follows:

decision variables	description
$X_{jik} \in \{0,1\}$	is 1 iff warehouse j supplies customer i with product k
$Y_j \in \{0,1\}$	is 1 iff warehouse is opened at location j
$U_{\ell jk} \geq 0$	shipment from plant ℓ to warehouse j of product k

The overall problem may now be stated:

$$
\left[\begin{array}{llcll}
\min & \displaystyle\sum_{j,k,\ell} c_{\ell jk} U_{\ell jk} & & & \text{plant-warehouse shipping costs} \\
+ & \displaystyle\sum_{i,j,k} d_{jik} w_{ik} X_{jik} & & & \text{warehouse-customer shipping costs} \\
+ & \displaystyle\sum_{j} f_j Y_j & & & \text{warehouse fixed costs} \\
\text{s.t.} & \displaystyle\sum_{j} X_{jik} & = & 1 & \text{single supplier} \\
& \displaystyle\sum_{i,k} s_k w_{ik} X_{jik} & \leq & q_j Y_j & \text{warehouse capacity} \\
& \displaystyle\sum_{i} w_{ik} X_{jik} & = & \displaystyle\sum_{\ell} U_{\ell jk} & \text{conservation at warehouses} \\
& \displaystyle\sum_{j} U_{\ell jk} & \leq & v_{\ell k} & \text{plant capacity} \\
& \displaystyle\sum_{j} Y_j & = & W & \text{fixed number of warehouses} \\
& X_{jik} & \in & \{0,1\} & \\
& Y_j & \in & \{0,1\} & \\
& U_{\ell jk} & \geq & 0 &
\end{array}\right]
$$

3 A Test Problem Set

Using a *factorial design* approach, we set min and max values (see Table 1) for the 5 design parameters: plants, warehouses, open warehouses, customers, and products. This resulted in $2^5 = 32$ problem sets. After randomly generating the data for each problem, we froze the data so that various solution strategies could be compared for each problem.

	plants	warehouses	open warehouses	customers	products
minimum	5	30	10	50	3
maximum	10	100	30	200	10

Table 1: Max and min values of design parameters.

4 CPLEX results

We first tried solving the problems in the test set using the CPLEX branch and bound (BB) code. The results of 1- and 2-hour CPLEX runs (see Table 2) verify that standard BB (or branch-and-cut, since CPLEX also considers a large variety of cuts with the default settings) does not in general produce satisfactory results for large-scale supply chain design problems. (We also tried a variety of strategy options with CPLEX and made longer runs, but these variants led to little improvement in the results.)

The following describe the labels in the table:

- CPLEX calculates % gap as $100(1 - \text{LB}/\text{UB})$, where UB is the value of the best feasible (integer) solution generated .
- OP means optimality after the indicated number of seconds. (Note that the CPLEX default setting for declaring optimality is a gap $\leq 0.01\%$.)
- NFSTL means CPLEX found no feasible solution within the time limit.
- ** means that the objective value did not improve in hour 2 of the run.

More specifically, these results (and results for additional problems of this type that we also considered) show that most of these problems are quite difficult for CPLEX, in the sense that for 21 of the 32 test problems it terminates either with a large optimality gap or with no feasible solution at all (the latter outcome occurs in 9 of the 32 problems). Thus, a method that takes advantage of problem structure is clearly necessary, and we therefore discuss below a Lagrangian approach that is widely used for supply chain design as well as a nested partition implementation. The results given below for these alternative approaches that take into account the structure of the supply chain problem demonstrate that the Lagrangian approach is sometimes successful for those cases in which CPLEX yields unsatisfactory results, but nested partitions is able to reliably generate higher quality solutions in a much shorter time than either CPLEX or Lagrange relaxation.

5 Lagrangian relaxation

Many hard integer programming problems can be viewed as easy problems complicated by a relatively small set of *side constraints*. Dualizing the side constraints produces a Lagrangian problem that is easy to solve and whose

optimal value is a *lower bound* (for minimization problems) on the optimal value of the original problem. The Lagrangian problem can thus be used in place of a linear programming relaxation to provide bounds in a branch and bound (BB) algorithm (Fisher 1981).The computational experience below demonstrates that a Lagrangian approach specialized to the supply chain problem yields significantly better lower and upper bounds than BB. (This approach was motived by a procedure described in (Pirkul and Jayaraman, 1996), but differs somewhat from their method in that the method described below relies more heavily on optimization tools, since we found some of the heuristics proposed by Pirkul and Jayaraman to be ineffective on our test set.) In particular, we relax single supplier constraints (with multipliers λ_{ik}) and conservation at warehouse constraints (with multipliers θ_{jk}). The resulting problem is

$$\begin{aligned} & \min \left(\sum_{j,k,\ell} c_{\ell jk} U_{\ell jk} + \sum_{i,j,k} d_{jik} w_{ik} X_{jik} + \sum_{j} f_j Y_j \right. \\ + & \left. \sum_{k,\ell} \theta_{jk} \left(\sum_{i} w_{ik} X_{jik} - \sum_{\ell} U_{\ell jk} \right) + \sum_{i,k} \lambda_{ik} \left(1 - \sum_{j} X_{jik} \right) \right), \end{aligned}$$

subject to the following constraints: warehouse capacity (X, Y variables), plant capacity (U variables), fixed number of warehouses (Y variables), and X, Y, binary variables.

Because the relaxation of conservation of flow decouples warehouses from plants, this problem can be decomposed into two separate problems P_1 and P_2. They are the following:

- Let Z_1 be the optimal value of the *plant-warehouse subproblem* P_1:

$$\left[\begin{array}{lll} \min & \displaystyle\sum_{j,k,\ell} (c_{\ell jk} - \theta_{jk}) U_{\ell jk} & \\ & \displaystyle\sum_{j} U_{\ell jk} & \leq \quad v_{\ell k} \\ & U_{\ell jk} & \geq \quad 0 \end{array} \right]$$

- Let Z_2 be the optimal value of the *warehouse-customer subproblem* P_2:

$$\left[\begin{array}{lll} \min & \displaystyle\sum_{i,j,k} (d_{jik} w_{ik} - \lambda_{ik} + \theta_{jk} w_{ik}) X_{jik} \quad + & \displaystyle\sum_{j} f_j Y_j \\ \text{s.t.} & \displaystyle\sum_{i,k} s_k w_{ik} X_{jik} & \leq \quad q_j Y_j \\ & \displaystyle\sum_{j} Y_j & = \quad W \\ & X_{jik} & \in \quad \{0,1\} \\ & Y_j & \in \quad \{0,1\} \end{array} \right]$$

It can shown that the warehouse-customer problem can also be decomposed into a separate set of warehouse problems, and the optimal solution to P_2 is then obtained by concatenating the best W solutions of these separate problems. Let $Z_{\lambda,\theta}$ be the optimal solution to this problem. Then $Z_{\lambda,\theta} = Z_1 + Z_2 + \sum_{i,k} \lambda_{ik}$ is a lower bound to the original problem. To maximize $Z_{\lambda,\theta}$, we use a subgradient algorithm.

prob	plants	warehouses	open	customers	products	% gap (1 hr – 2 hr)
1	5	30	10	50	3	720 OP
2	5	30	10	50	10	16.72 % – 15.05 %**
3	5	30	10	200	3	7.64 % – 5.45%
4	5	30	10	200	10	NFSTL
5	5	30	20	50	3	28 OP
6	5	30	20	50	10	0.12 % – 0.04 %
7	5	30	20	200	3	1200 OP
8	5	30	20	200	10	8.29 %–7.61 %**
9	5	100	10	50	3	6.89 % – 3.52 %
10	5	100	10	50	10	NFSTL – 25.42 %
11	5	100	10	200	3	NFSTL
12	5	100	10	200	10	NFSTL
13	5	100	20	50	3	2900 OP
14	5	100	20	50	10	15.29 % – 15.29 %**
15	5	100	20	200	3	11.69 % – 11.11 %**
16	5	100	20	200	10	NFSTL
17	10	30	10	50	3	1100 OP
18	10	30	10	50	10	22.03 % – 18.47 %**
19	10	30	10	200	3	18.30 % – 16.82 %**
20	10	30	10	200	10	NFSTL
21	10	30	20	50	3	160 OP
22	10	30	20	50	10	3.09 % – 0.09 %
23	10	30	20	200	3	0.75 % – 0.12 %
24	10	30	20	200	10	7.02 % – 6.57 %**
25	10	100	10	50	3	8.27 % – 6.64 %**
26	10	100	10	50	10	NFSTL
27	10	100	10	200	3	NFSTL
28	10	100	10	200	10	NFSTL
29	10	100	20	50	3	6.99 % – 0.21 %
30	10	100	20	50	10	19.64 % – 19.32 %**
31	10	100	20	200	3	11.63 % – 11.35 %**
32	10	100	20	200	10	NFSTL

Table 2: Gaps for 1- and 2-hour CPLEX runs (using default CPLEX settings).

6 Lagrange/CPLEX results

Combining the Lagrangian relaxation process described above with a feasibility process (submission of "best" W warehouses as ranked by the Lagrangian process to CPLEX for completion of best feasible solution using exactly those W warehouses), we obtained results that in many cases reflected significant improvements relative to "pure" CPLEX runs. These are summarized in Table 3. Note, however, that a number of problems are difficult for both pure

	Lagrange/CPLEX			CPLEX		
prob	LB	objective value	% gap	LB	objective value	% gap
1	352,530	366,541	3.82%	364,550	364,588	0.01%
2	1,269,931	1,350,625	5.97%	1,216,545	1,432,145	15.05%
3	1,542,962	1,590,121	2.97%	1,530,531	1,618,810	5.45%
4	5,072,828	5,421,684	6.43%	-	-	NFSTL
5	323,426	337,800	4.26%	336,421	336,445	0.01%
6	1,073,532	1,102,424	2.62%	1,097,614	1,098,013	0.04%
7	1,344,637	1,386,138	2.99%	1,385,340	1,385,480	0.01%
8	4,251,166	4,386,238	3.08%	4,192,110	4,537,556	7.61%
9	296,621	325,681	8.92%	310,448	321,776	3.52%
10	1,134,625	1,293,571	12.29%	1,044,617	1,400,750	25.42%
11	1,223,625	1,339,852	8.67%	-	-	NFSTL
12	4,069,607	5,361,418	24.09%	-	-	NFSTL
13	243,840	253,623	3.86%	253,209	253,235	0.01%
14	927,728	992,490	6.53%	881,018	1,040,055	15.29%
15	1,007,195	1,073,615	6.19%	990,423	1,114,251	11.11%
16	2,948,580	4,217,126	30.08%	-	-	NFSTL
17	305,592	330,420	7.51%	325,031	325,109	0.02%
18	1,089,266	1,158,089	5.94%	1,037,379	1,272,316	18.47%
19	1,298,565	1,372,471	5.38%	1,247,911	1,500,260	16.82%
20	4,355,511	4,875,456	10.66%	-	-	NFSTL
21	263,113	273,176	3.68%	272,617	272,658	0.02%
22	920,990	952,316	3.29%	947,667	948,531	0.09%
23	1,115,479	1,156,966	3.59%	1,151,281	1,152,714	0.12%
24	3,649,089	3,848,794	5.19%	3,635,487	3,891,324	6.57%
25	291,743	323,563	9.83%	308,270	330,181	6.64%
26	956,546	1,112,019	13.98%	-	-	NFSTL
27	1,106,097	1,256,834	11.99%	-	-	NFSTL
28	3,390,842	4,624,505	26.68%	-	-	NFSTL
29	245,319	265,072	7.45%	256,502	257,032	0.21%
30	771,270	865,677	10.91%	753,794	934,290	19.32%
31	889,280	964,746	7.82%	857,597	967,425	11.35%
32	2,391,263	3,592,618	33.44%	-	-	NFSTL

Table 3: Comparison of 2-hour Lagrange/CPLEX runs and 2-hour CPLEX runs.

CPLEX and our L-CPLEX hybrid approach. For these problems (such as 12, 16, 28, 32), we demonstrate below that an nested partitions approach provides a vehicle for obtaining better feasible solutions more rapidly and also provides evidence (via sampling) that the lower bounds generated by pure CPLEX and by the L-CPLEX hybrid are unrealistic in the sense that the distribution of objective values of randomly generated solutions suggests that those lower bounds are well below any objective value that is likely to be obtained.

7 Nested partitions (NP)

We first sketch a generic nested partition procedure and then describe an implementation of NP that takes advantage of the structure of the supply chain problem.

Consider the following combinatorial optimization problem, where Θ is a finite

set: $\min_{x \in \Theta} f(x)$. The *nested partitions* (NP) framework can be briefly described as follows. In each iteration of the algorithm we assume that we have a region (subset) of Θ that is considered *the most promising* (initially, this is Θ). We then partition this most promising region into subregions (for example, M subregions) and aggregate the other regions (termed the complementary region) into one region. At each iteration, we thus consider $M + 1$ disjoint subsets of the feasible region Θ. Each of these $M + 1$ regions is sampled using some random sampling scheme, and for each region a *promise index* is calculated. (A variety of approaches may be used to calculate the promise index see (Shi and Olafsson 2000), but in the results below we simply used the objective value of tbe best sample in the region. Other possibilities in this context would involve functions that took into account both the best sample, which provides an upper bound for the region, and the LP relaxation value, which provides a lower bound for the region.) These promise indices are then compared to determine which region is the most promising in the next iteration. If one of the child subregions is found to have the best promise index, this subregion becomes the most promising region. (In an ideal instance, an unbroken sequence of child regions containing an optimal solution would be obtained, yielding a depth-first search that quickly leads to an optimal solution.) If, on the other hand, the surrounding region is found to have the best promise index, a subset of the surrounding region becomes the most promising. The new most promising region is then partitioned and sampled in a similar fashion. Since Θ is finite, in a finite number of iterations we reach regions that contain only a single point, and for such a region the sampling procedure generates the single feasible point, and no further consideration of that region is required. As the algorithm evolves, a sequence of most promising regions $\sigma(k)$ will be generated, where $\sigma(k)$ is the most promising region in the k-th iteration. In (Shi and Ólafsson 2000), we have shown that $\sigma(k)$ is a *Markov* chain with all the global optima as its absorbing states. Thus, global convergence is guaranteed. The only requirement on the random sampling procedure is that each point in the region has a positive probability of being selected. Hence there is much flexibility in selecting a sampling procedure.

In order to take advantage of the supply chain problem structure via NP, we first construct a heuristic as described below to develop a set of efficiency indices for the warehouses. These indices provide a ranking of warehouses that is then used as a basis for the selection of a subset of the warehouses as a "warm start" promising region for NP. The ranking is then also used as a basis for biased sampling approaches for an NP implementation that uses the AMPL modeling language in conjunction with CPLEX to generate subproblems for the promising region and complementary region and to evaluate the samples for these regions.

7.1 A Heuristic Procedure for Ranking Warehouses

We generate via the following three step procedure a heuristic warehouse ranking based on average unit costs for the warehouses by solving (via CPLEX) an optimization problem for each warehouse and then scaling the optimal value by dividing by the warehouse capacity :

Warehouse Ranking

1: Pick a warehouse from the set of warehouses and open it . Close all other warehouses. Solve the problem that requires that the total warehouse throughput equal the warehouse capacity, replacing the customers' "demand" equations by upper bounds that limit product-customer shipments to at most the demand. (The integer variables corresponding to product-customer pairs are relaxed to continuous variables.) The optimal value of this linear programming problem (involving only one warehouse but all plants and customers) yields the lowest total cost associated with full utilization of the warehouse. (Note that this optimal value also includes the fixed cost of the warehouse.)

2: Calculate an average unit cost for each warehouse by the dividing the total cost by the (fully utilized) capacity of the warehouse.

3: Define an efficiency ranking for the warehouse according to the average unit costs.

(We are currently experimenting with variants of this ranking process in which the throughput is set to an appropriate fraction of the warehouse capacity.)

7.2 An NP Implementation

We now describe an NP implementation that uses these unit cost rankings to generate a "warm start" followed by biased sampling guided by the global NP viewpoint. This approach involves the following parameters and sets:

- Select the top R ranked warehouses and place them in a set called TOP, and place the remaining warehouses are put into a set called BOTTOM.
- p: The probability of choosing a warehouse from TOP.
- initNP: The number of warehouses to be set as open at the starting point of each major iteration. Here a major iteration is defined as the branching process from the starting point to the first node at which all warehouses are fixed (at which point a backtrack step is made to a new starting point).

- S: The number of samples to be generated in each region (including the complementary regions). Each sample is a set of W open warehouses in the feasible set of the corresponding region, and the optimal cost of the supply chain network with this set of warehouses is determined by CPLEX. (In this context, each sample requires about 3 seconds to evaluate on a 500MHz computer.)

NP Implementation for Supply Chain Design

The NP implementation is as follows:

1. If iteration = 1 then the initial promising region is obtained by fixing the highest ranked initNP warehouses as open. Else obtain a promising region by fixing an appropriate subset of BestOPEN (see below) as open.
2. Generate S samples for each of the following regions:
a. Region 1 (first subset of the promising region): another warehouse is selected and set as open (this warehouse is randomly selected from the set TOP with probability p, and from the set BOTTOM with probability (1-p))

b. Region 2 (remainder of the promising region): the newly selected warehouse is closed

c. Region 3: complementary region (at least one of the warehouses used to define the promising region is not allowed in the warehouse sets considered in this region)

For each region the remaining warehouses needed to complete each sample of W warehouses are chosen from TOP and BOTTOM, where the probability of picking a warehouse from TOP is p and picking a warehouse from BOTTOM is (1-p). CPLEX is used to find the optimal shipping pattern for each sample (set of warehouses). NP keeps track of the best sample as BestOPEN and uses this information to either continue the path down from the promising region or to backtrack to a subet of the complementary region.

The remainder of the process is analogous to the generic NP procedure. If backtracking is needed because the bottom of the tree is reached (all warehouses are fixed) or because the best sample occurs in the surrounding region, then we choose initNP warehouses from BestOPEN as the new starting set (performing the selection so that this set does not coincide with a previous starting set) and return to step 1. In summary, NP utilizes the results of the warehouse ranking heuristic to begin a search process that is initially focused around the warehouse sets predicted to be in the optimal solution according to this ranking, but nevertheless maintains a global view of the feasible set and therefore allows completely different solutions (unrelated to the ranking) to be considered.

The results in table Table 4 show that the NP strategy is very effective in using this heuristic information. Our testing focused on the four hardest problems in the set, for which NP obtained higher quality solutions than either

CPLEX (which was not able to obtain any feasible solutions for any of these four problems) or Lagrange relaxation. Moreover, the solution times required by NP in this context are roughly 1/3 of the 2-hour times used by CPLEX and Lagrangian relaxation for each of the problems.

problem	CPLEX	Lagrange	nested partitions
12	NFSTL	5,361,418	5,176,560
16	NFSTL	4,217,126	4,161,450
28	NFSTL	4,624,505	4,539,630
32	NFSTL	3,592,618	3,251,090

Table 4: Comparison of the three methods on the hardest problems

8 Conclusions and Directions for Further Research

Our computational results demonstrate that for large-scale supply chain design problems, the nested partitions approach can substantially outperform both general purpose combinatorial optimizers (such the branch-and-cut solver within CPLEX) and specialized approaches such as Lagrangian relaxation. Nested partitions can effectively utilize problem-specific heuristics that are difficult to incorporate within other combinatorial optimization approaches, and for this problem class we have developed an excellent warehouse ranking heuristic and used this heuristic to construct a "warm start" procedure followed by an effective biased sampling approach that is guided by a global view of the problem. This research has also established the applicability of the AMPL/CPLEX modeling-language/solver combination as a vehicle for the implementation of global meta-heuristics, and this represents a novel and successful use of these powerful software tools.

This research also opens many directions for further investigation. One particularly promising direction that we are pursuing involves the construction of composite lists of preferred warehouses. In the results above, we used biased sampling based upon a split of the warehouse list into a TOP group of warehouses ranked highly by the average efficiency index, and a set BOTTOM with the remaining warehouses. Rather than simply relying on one ranking scheme such as the one corresponding to efficiency, we have also considered more sophisticated approaches such as composite TOP groups comprised of warehouses selected from two different ranking schemes (for example, combining warehouses ranked highly by either efficiency or capacity has produced excellent results in some problems). Alternatively, the TOP group could evolve dynamically by adding to it new warehouses that appear in high-quality solutions. Finally, as suggested above, the promise index for a region could be made a function of both lower bound (relaxation) and upper

bound (sampling) information, since both types of information are available within this hybrid approach.

REFERENCES

1. Shi, L. and S. Olafsson (2000), "Nested Partitions Method for Global Optimization", Operations Research, 48 ,390-407.

2. Shi, Leyuan, Sigurdur Olafsson, and Qun Chen (1999), "A New Hybrid Optimization Algorithm", Computers and Industrial Engineering, 36, 409-426.

3. Pirkul,H. and V. Jayaraman (1996), "Production, Transportation, and Distribution Planning in a Multi-Commodity Tri-Echelon System" Transportation Sciences,30, No. 4, 291-302.

4. M. Fisher (1981)., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", Management Sciences, 27, 1-18