Solving the Concave Cost Supply Scheduling Problem

Xia Wang, ***Univ. of Maryland***

Bruce Golden, ***Univ. of Maryland***

Edward Wasil, ***American Univ.***

Presented at EURO XXI

Iceland, July 2006

Outline of Lecture

- Introduce the supply scheduling problem
- Discuss the two cases from the literature
- Formulate the problem mathematically
- Present and illustrate solution procedures
- Computational results
- Future work

Problem Statement: Case One

- Based on work by Chauhan &Proth, EJOR, 2003 and Chauhan et al., ORL, 2005
- There are n providers and one manufacturing unit
- The manufacturing unit has a demand of D
- The capacity of provider i is M_i
- If there is positive flow x_i from provider i to the manufacturing unit, then $m_i \le x_i \le M_i$
- The mathematical formulation is given next

Case One Formulation

- Short-hand formulation

$$Min \quad Z = \sum_{i=1}^{n} c_i(x_i)$$

$$s.t. \quad \sum_{i=1}^{n} x_i = D$$

$$x_i \in \{0\} \cup [m_i, M_i] \qquad i = 1, 2, \ldots, n$$

- The cost function

$$c_i(x) = \begin{cases} a_i + b_i(x) & x > 0 \\ 0 & x = 0 \end{cases}$$

where $a_i \geq 0, \lim_{x \to 0^+} b_i(x) \geq 0$
and $b_i(x)$ is concave, continuously differentiable, and increasing

Problem Statement: Case Two

- There are n providers and m manufacturing units
- Manufacturing unit j has a demand of D_j
- The capacity of provider i is M_i
- If there is positive flow x_{ij} from provider i to manufacturing unit j, then $m_{ij} \leq x_{ij} \leq M_i$
- The mathematical formulation is given next

Case Two Formulation

- Short-hand formulation

$$Min \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}(x_{ij})$$

$$s.t. \quad \sum_{j=1}^{m} x_{ij} \le M_i \qquad i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = D_j \qquad j = 1,2,\ldots,m$$

$$x_{ij} \in \{0\} \cup [m_{ij}, M_i] \qquad i = 1,2,\ldots,n,\ j = 1,2,\ldots,m$$

- The cost function

$$c_{ij}(x) = \begin{cases} a_{ij} + b_{ij}(x) & x > 0 \\ 0 & x = 0 \end{cases}$$

where $a_{ij} \ge 0,\ \lim_{x \to 0^+} b_{ij}(x) \ge 0$
and $b_{ij}(x)$ is concave, continuously differentiable, and increasing

Solution Procedure Overview

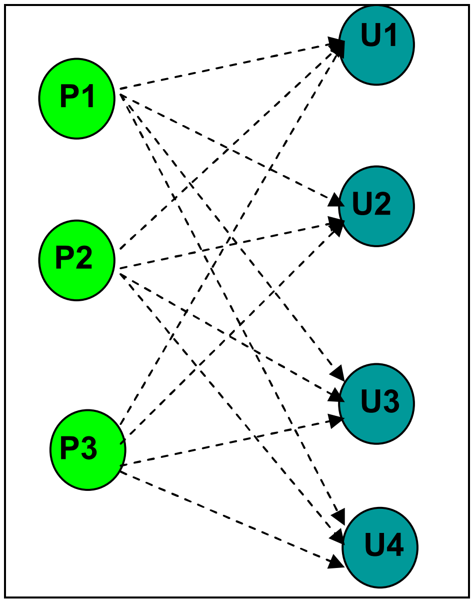
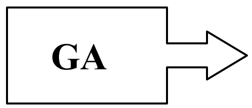


Figure 1



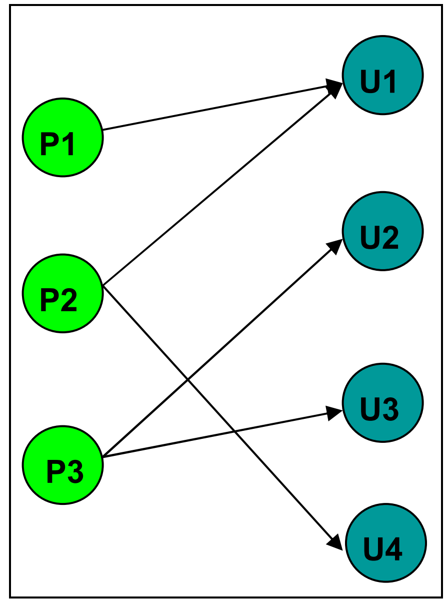
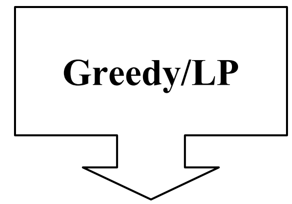


Figure 2



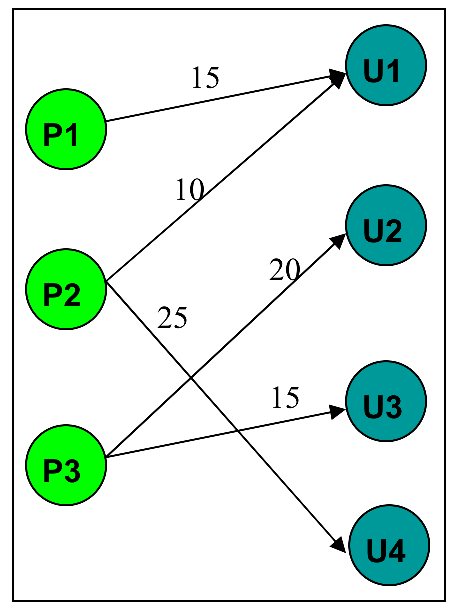


Figure 3

Case One Genetic Algorithm (GA1)

- Initialization
  - Randomly generate a $POP \times n$ binary matrix

$$P_{POP,n} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}$$

    where POP is the population size in GA1

  - We used POP = 50
  - Each row is a chromosome

GA1 Continued

- Feasibility check

  For each chromosome (e.g., row k), we check whether

$$\sum_{i=1}^{n} M_i \, P_{ki} \geq D$$

  If not, we randomly turn a gene (an entry in row k) from 0 to 1 until the constraint is satisfied

- Phase 1. Select the providers
  - Crossover using Queen-Bee Selection (two-point crossover)
  - Mutation (one flip per chromosome)
  - Feasibility check

GA1 Continued

- Phase 2. Assignment of flows
  - 2.1 Greedy assignment for each offspring

Step 1: For each provider, determine the maximum quantity Q_i that he can provide

Step 2: Select the provider (i) with the smallest average cost per unit, given that he ships Q_i to the manufacturing unit

Step 3: Update the remaining demand and the capacity of provider i

Step 4: Go to Step 1, unless the demand is fully satisfied

Step 5: Calculate the total cost for each offspring

GA1 Continued

- 2.2 Population selection/survival

  Given 50 parents and 50 children (after mutation), keep the best chromosome and select the remaining population for the next generation using proportional selection

- Phase 3. Terminate or continue
  - If the best solution has remained the best for five consecutive generations, then terminate
  - Otherwise, return to Phase 1

Computational Results for Case One

- Results are presented by Chauhan & Proth, EJOR, 2003
- We compare GA1 with CP on 20 benchmark problems
- There are six providers and the variable cost functions are concave
- GA1 beats CP on 6 and 14 ties
- All 20 GA1 solutions are optimal (mention GA2)
- GA1 running times are less than a second

Case Two Genetic Algorithm (GA2)

- Initialization
  - Repeat Steps 1 - 4 50 times to obtain an initial population

Step 1: Choose a manufacturing unit at random

Step 2: Apply the greedy method discussed earlier (Phase 2, GA1) that uses the notion of smallest average cost per unit to satisfy demand at that manufacturing unit

Step 3: Pick another manufacturing unit and go to Step 2 until the demands of all manufacturing units have been satisfied

Step 4: $P_{ij} = 1$ if provider *i* has been assigned to manufacturing unit *j* and $P_{ij} = 0$ otherwise

GA2 Continued

- At this point, we have an initial population
- Each chromosome is an $n \times m$ binary matrix
- Each chromosome admits at least one feasible solution
- Since we want to compare the results of GA2 with optimal solutions, we only consider linear variable costs as shown below.

$$c_{ij}(x) = \begin{cases} a_{ij} + b_{ij}x & x > 0 \\ 0 & x = 0 \end{cases}$$

GA2 Continued

- For each chromosome in the population, we obtain a fitness (or cost) by solving the LP below (in MATLAB)

$$Min \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} P_{ij} + b_{ij} x_{ij})$$

$$s.t. \quad \sum_{j=1}^{m} x_{ij} \le M_i \qquad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = D_j \qquad j = 1, 2, \ldots, m$$

$$m_{ij} P_{ij} \le x_{ij} \le M_i P_{ij} \qquad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m$$

GA2 Continued

- Phase 1. Select the edges in the graph
  - Crossover
  - Mutation
- Phase 2. LP and assignment of flows
  - Given the edges for each offspring, use the LP model to obtain flows and total cost
  - Given 50 parents and 50 children, keep the best individual and select the remaining 49 for the next generation using proportional selection

GA2 Continued

- Phase 3. Terminate or continue
  - If the best solution has remained the best for five consecutive generations, then terminate
  - Otherwise, return to Phase 1
- An illustration of GA2
  - 3 providers and 4 manufacturing units
  - In the initialization step, we generate 50 chromosomes like the one on the next page

An Illustration of GA2

- A chromosome

$$P = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

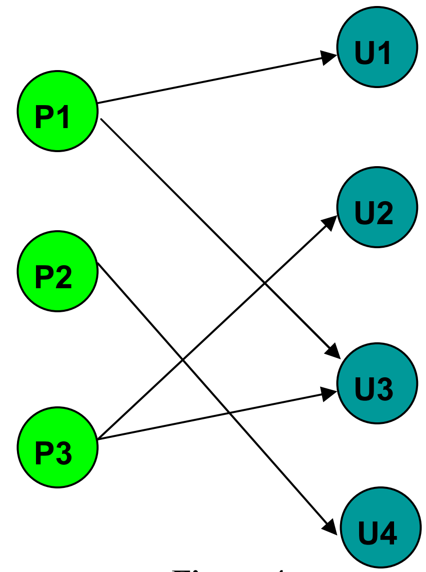


Figure 4

- For every chromosome in the initial population, we solve the LP model to obtain its fitness (i.e., total cost)

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{\text{LP}} \begin{pmatrix} 15 & 0 & 10 & 0 \\ 0 & 0 & 0 & 25 \\ 0 & 20 & 15 & 0 \end{pmatrix}$$

An Illustration of GA2

- The resulting flow network is at right
- In Phase 1, we apply Queen-Bee crossover
- The best solution is the Queen-Bee

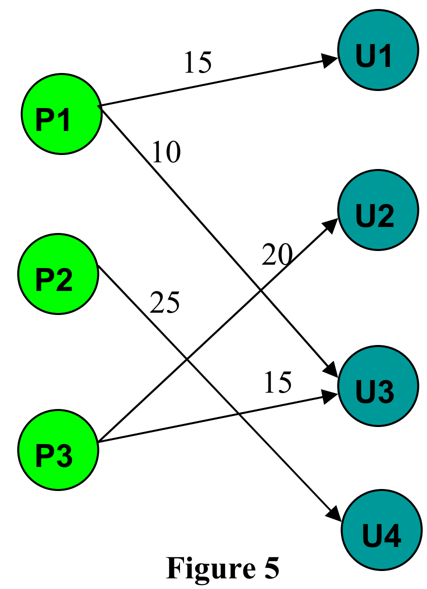


Figure 5

- We select 25 other individuals in proportion to their fitness to mate with the Queen
  - In crossover, we open the edge from provider *i* to unit *j* if it is open in the Queen or the other individual
  - Otherwise, the edge is not opened

An Illustration of GA2

- Crossover between the Queen and another individual is shown below

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cup \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

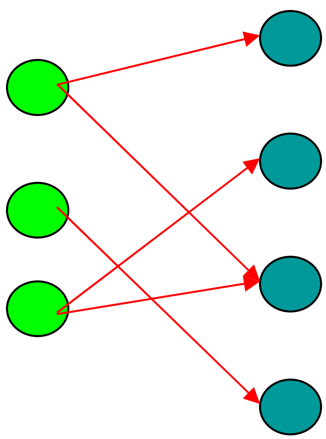


Figure 6

$\cup$

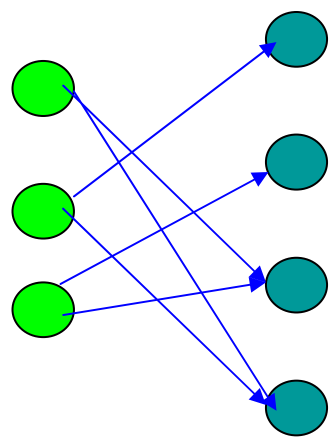


Figure 7

$\Rightarrow$

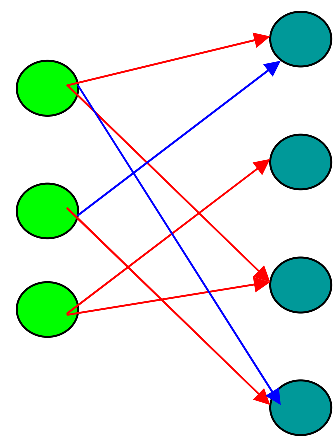


Figure 8

An Illustration of GA2

- After taking the union of edges, we apply the greedy method to eliminate redundant edges (see below)
- This completes crossover

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{\text{Greedy Method}} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

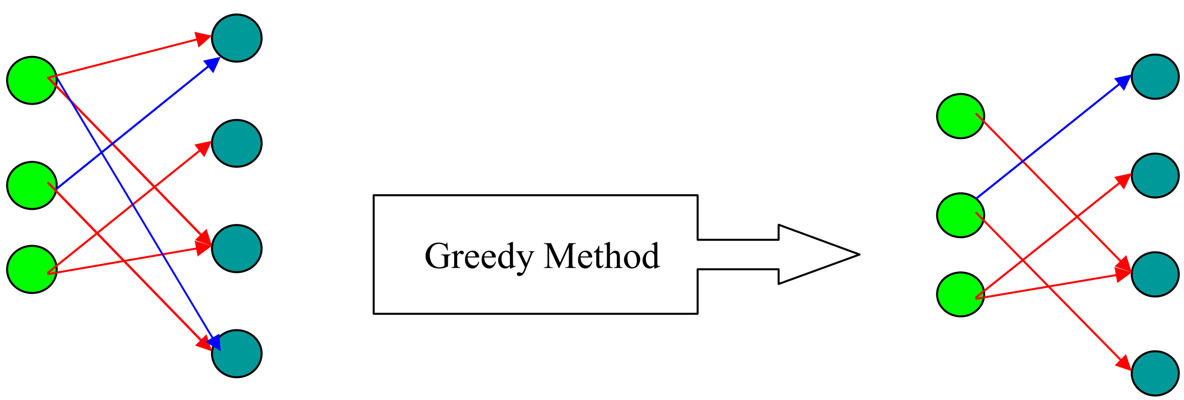


Figure 9 **Figure 10**

An Illustration of GA2

- The mutation operator is shown below
- This completes Phase 1

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} \underline{0} & 1 & 1 & 1 \\ \underline{1} & 1 & 1 & 1 \\ \underline{1} & 1 & 1 & 1 \end{pmatrix} \xrightarrow{\text{Greedy Method}} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

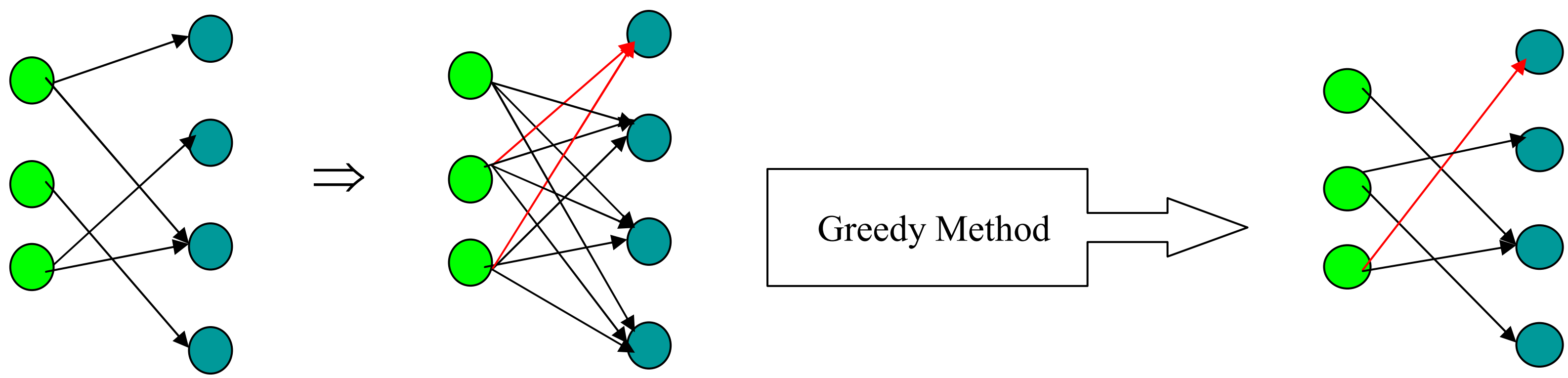


Figure 11 Figure 12 Figure 13

An Illustration of GA2

- We now have 25 children from crossover and 25 children from mutation
- We use LP to determine the cost of each child
- We determine the next generation, as indicated earlier
- This completes Phase 2
- In Phase 3, we continue until the termination rule is satisfied
- Optimal solutions to the MIP are obtained using Xpress
- The MIP model is shown on the next page

MIP Formulation for Case Two

$$Min \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} y_{ij} + b_{ij} x_{ij})$$

$$s.t. \quad \sum_{j=1}^{m} x_{ij} \le M_i \qquad i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = D_j \qquad j = 1,2,\ldots,m$$

$$m_{ij} y_{ij} \le x_{ij} \le M_i y_{ij} \qquad i = 1,2,\ldots,n, \; j = 1,2,\ldots,m$$

$$x_{ij} \ge 0 \text{ and } y_{ij} \in \{0,1\}$$

- GA2 is coded in MATLAB
- The experiments are run on a 3.06 GHz Pentium IV machine with 1GB RAM

Computational Results

Case	Method	Cost	Gap(%)	Time (Sec.)
10P12U	GA	336	0.00	147.2
	OPT	336	–	0.7
12P15U	GA	401	1.01	128.7
	OPT	397	–	1.1
15P18U	GA	507	0.00	256.0
	OPT	507	–	0.9
18P20U	GA	514	0.00	277.3
	OPT	514	–	6.6
20P25U	GA	551	0.92	208.2
	OPT	546	–	4.4

Case	Method	Cost	Gap(%)	Time (Sec.)
25P30U	GA	620	1.47	323.6
	OPT	611	–	69.1
30P35U	GA	805	2.16	432.5
	OPT	788	–	3604.1
40P50U	GA	745	5.08	452.8
	OPT	709	–	5045.2
45P50U	GA	845	5.10	555.9
	OPT	804	–	6195.7
50P55U	GA	876	4.66	610.2
	OPT	837	–	36606.3

- GA cost is best of 10 replications
- GA time (sec.) is the total over 10 replications

A Test of Robustness

- We might be concerned about an unexpected increase in demand
- If so, we can easily add an LP constraint within GA2 such as
  - The capacity of available suppliers should be large enough to handle a pre-specified demand increase at the manufacturing units
- For example, we might plan for a demand increase of up to 3% at each manufacturing unit
- GA3 is the solution with this added constraint
- GA0 is the solution without it

Test of Robustness

Problem	Method	Cost	Gap(%)	Time (sec.)
10P12U	OPT	336	–	0.7
	GA0	336	0.00	147.2
	GA3	336	0.00	180.7
12P15U	OPT	397	–	1.1
	GA0	401	1.01	128.7
	GA3	401	1.01	198.8
15P18U	OPT	507	–	0.9
	GA0	507	0.00	256.0
	GA3	510	0.59	300.9
18P20U	OPT	514	–	6.6
	GA0	514	0.00	277.3
	GA3	516	0.39	470.2
20P25U	OPT	546	–	4.4
	GA0	551	0.92	208.2
	GA3	548	0.37	265.1

Problem	Method	Cost	Gap(%)	Time (sec.)
25P30U	OPT	611	–	69.1
	GA0	620	1.47	323.6
	GA3	626	2.45	391.7
30P35U	OPT	788	–	3604.1
	GA0	805	2.16	432.5
	GA3	817	3.68	477.9
40P50U	OPT	709	–	5045.2
	GA0	745	5.08	452.8
	GA3	745	5.08	478.5
45P50U	OPT	804	–	6195.7
	GA0	845	5.10	555.9
	GA3	843	4.85	600.0
50P55U	OPT	837	–	36606.3
	GA0	876	4.66	610.2
	GA3	891	6.45	720.2

- The cost of a more robust solution is very small

Conclusions and Future Work

- The GA is relatively simple, quick, and powerful
- The GA solutions are within 5% of optimality
- The GA can easily find robust solutions
- The incremental cost is small
- We think we can do a little better
- This will be the focus of future work