Journal of the Korean Institute of Industrial Engineers
Vol. 15, No. 2, Dec., 1989

Machine Repair Problem in Multistage Systems†

Young-Taek Park

직렬시스템의 수리 및 예비품 지원정책에 관한 연구

박 영 택*

Abstract

The classic machine repair problem is extended to the case where a number of different machines are arranged in the sequence of operation. The steady-state availability of the system with a series of operating machines is maximized under some constraints such as total cost, available space. In order to find the optimal numbers of spare units and repair channels for each operating machine, the problem is formulated as non-linear integer programming(NLIP) problem and an efficient algorithm, which is a natural extension of the new Lawler-Bell algorithm of Sasaki et al., is used to solve the NLIP problem. A numerical example is given to illustrate the algorithm.

1. Introduction

In the classic machine repair problem or repairman problem[1, 2], m identical machines are operating in parallel, and s spare machines and a repair facility having x parallel repair channels(i.e., capable of repairing x machines simultaneously; obviously, we could consider the facility as consisting of x repairmen) support the operating machines. A problem in the machine repair with spares model is the determination of the optimal numbers of spare units and repair channels[10, 11]. The classic

* Department of Industrial Engineering, Sung-Kyun-Kwan University
† This research was supported by Korea Science & Engineering Foundation

machine repair problem has been treated and extended extensively in the literature. Some of them are the cases of a multi-stage repair[5], heterogeneous machines[6], and multiple repair levels[7].

In the classic machine repair problem and its extensions, there is only one operating stage and all the operating machines are doing the same job. However,in many cases, the number of operating stages is more than one and the operating machine at each stage is doing its own job. A series of operating machines, arranged in the sequence of operation, in a production line may be the case.

This paper deals with a machine repair problem for a series of operating machines. The steady-state availability of the system with a series of operating machines is maximized under some constraints such as total cost, available space, etc.

Sasaki et al.[9] considered a similar problem, but in their study a single repair channel was provided for each operating machine and the problem was only to determine the number of spare units for each operating machine. The purpose of this paper is to determine simultaneously the optimal number of repair channels and the optimal number of spare units for each operating machine. The problem is formulated as a constrained NLIP problem and the new Lawler-Bell algorithm proposed by Sasaki et al.[9] is tailored to solve the problem efficiently. A numerical example is given to illustrate the optimization procedure.

2. Model Formulation

The following assumptions and notations are used in this paper.

2-1. Assumptions

1. In each operating stage, a machine is operating and a number of spare units and repair channels support the operating machine.
2. The operating stages are connected in series.
3. The distributions of lifetime and repair time at each stage are exponential.
4. The time for replacement is negligible.
5. The failures of units in the system are mutually independent.
6. Spare units in standby do not fail.

2-2. Notations

n : number of operating stages in the system

x_j : number of repair channels at j-th stage

: number of total units(operating unit plus spare units) at j-th stage

r_j : (failure rate/repair rate) of a unit used in j-th stage

$A_j(x_j,y_j)$: steady-state availability of j-th stage

c_{1j},c_{2j} : cost of a repair channel and a unit at j-th stage

s_{1j}, s_{2j} : required spaces of a repair channel and a unit at j-th stage,

C : upper bound of system cost

S : total available space

Other notations are defined as needed.

Using the steady-state probabilities of the classical machine repair problem[4], we obtain the following steady-state availability of j-th stage:

$$A_j(x,y)=\begin{cases}\dfrac{\Sigma_{k=0}^{x-1}(r_j^k/k!)+\Sigma_{k=x}^{y-1}(r_j^k/x^{k-x}x!)}{\Sigma_{k=0}^{x-1}(r_j^k/k!)+\Sigma_{k=x}^{y}(r_j^k/x^{k-x}x!)} & \text{, if } x\le y-1 \\ \Sigma_{k=0}^{y-1}(r_j^k/k!)/\Sigma_{k=0}^{y}(r_j^k/k!) & \text{, if } x\ge y. \end{cases} \quad \cdots\cdots (1)$$

Since the operating stages are connected in series, and the number of repair channels need not be larger than the number of units in each stage, the problem is formulated as the following NLIP problem:

$$\text{Maximize} \quad \Pi_{j=1}^{n} A_j(x_j, y_j) \cdots\cdots (2.1)$$

subject to

$$\Sigma_{j=1}^{n} (c_{1j}x_j+c_{2j}y_j)\le C \cdots (2.2)$$

$$\Sigma_{j=1}^{n} (s_{1j}x_j+s_{2j}y_j)\le S \cdots (2.3)$$

$$x_j\le y_j(;\ j=1,2,\cdots,n) \cdots (2.4)$$

x_j and y_j are nonnegative integers,

where $A_j(x_j, y_j)$ is given by equation(1).

Equation(2.1) is the steady-state probability that all the n stages are simultaneously operating where their individual operations are independent, i.e., nonfailed stages continue their operations during the repair of a failed stage. If all the other stages stop their operations during the repair of a failed stage, equation(2.1) understates the system availability. See details in Fox and Zerbe[3].

3. Solution Procedure

In the previous section, the problem is formulated as a NLIP problem. To solve a NLIP problem by the Lawler-Bell(LB) algorithm, all the variables must be expanded to binary ones. To obviate this problem, Sasaki et al.[9] proposed the new Lawler-Bell(NLB) algorithm, which allows integer variables. Our problem can be solved by the LB or NLB algorithm, but the optimal solution can be obtained more efficiently by exploiting the special property of the problem.

3-1. Preliminaries

Consider any n-component vector $V=(v_1, v_2,\cdots,v_n)$, where $\underline{v}_j<v_j<\bar{v}_j$, and all the $\underline{v}_j$, v_j, $\bar{v}_j$ are nonnegative integers.

Definition 1: Numerical Ordering

The numerical ordering, N(V), of a vector V is defined as the following numerical value:

$$N(V)\equiv\Sigma_{j=1}^{n} v_j(\bar{v}+1)^{j-1} \text{ where } \bar{v}=\max\{\bar{v}_j\}.$$

N(V) can be interpreted as a size of V. V is defined larger in the numerical ordering than V′, if and only if $N(V)>N(V')$.

Proposition 1: $N(V)=N(V')$ if and only if $v_j=v'_j$ for all $j=1, 2,\cdots,n$, where $V'=(v'_1,$

$v'_2,\cdots,v'_n)$.

Definition 2: Vector Partial Ordering

V is defined larger in the vector partial ordering than V'(; $V \geq V'$) if and only if $v_j \geq v'_j$ for all $j=1, 2,\cdots,n$. Similarly, $V \leq V'$ if and only if $v_j \leq v'_j$ for all $j=1, 2,\cdots,n$.

Proposition 2: If a function, $f(V)$, is monotone nondecreasing in each of the variables $v_1, v_2,\cdots,v_n$, $V \geq V'$ implies $f(V) \geq f(V')$.

Definition 3: For any given vector V, three vectors V^*, V^+, and V^{++} are defined as:

$$V^* = \max_{N(V')}\{V' \mid N(V') < N(V),\ V' \nleq V\},$$

$$V \ = \min_{N(V')}\{V' \mid N(V^*) < N(V')\},$$

$$V^{++} = \max_{N(V')}\{V' \mid N(V') < N(V)\},$$

where "$\nleq$" is the negation of "$\leq$".

Suppose that all V's are listed in numerical descending order, i.e.,

$$V_{max} \equiv (\bar{v}_1, \bar{v}_2,\cdots,\bar{v}_n),$$
$$(\bar{v}-1, \bar{v}_2,\cdots,\bar{v}_n),$$
$$\vdots$$
$$V_{min} \equiv (\underline{v}_1, \underline{v}_2,\cdots,\underline{v}_n).$$

Then, V^* is the first vector following V in the numerical ordering that has the property $V^* \nleq V$; V^+ is the vector immediately before V^*; and V^{++} immediately follows V. V^*, V^+, V^{++} are obtained as follows.

Generation of V^*: find the minimum i that satisfies $v_i \doteqdot \bar{v}_i$. If $i=n$, or $v_j = \underline{v}_j$ for all $j > i$, then V^* cannot be defined. Otherwise, find the minimum i' that satisfies $i' > i$ and $v_{i'} \doteqdot \underline{v}_{i'}$. Then,

$$V^* = (\bar{v}_1,\cdots, \bar{v}_{i'-1}, \bar{v}_{i'-1}, v_{i'+1},\cdots,v_n).$$

Remark 1: If $V = V_{max}$, V^* cannot be defined since there exists no such i that satisfies $v_i \doteqdot \bar{v}_i$.

Remark 2: If $V = V_{min}$, V^* cannot be defined since there exists no such i' that satisfies $v_{i'} \doteqdot \underline{v}_{i'}$

Generation of V^+: If $V = V_{max}$, then $V^+ = V_{min}$. Otherwise, find the minimum i that satisfies $v_i \doteqdot \bar{v}_i$. Then,

$$V^+ = (\underline{v}_1,\cdots,\underline{v}_i, v_{i+1},\cdots,v_n).$$

Generation of V^{++}: Find the minimum i that satisfies $v_i \doteqdot \underline{v}_i$. Then,

$$V^{++} = (\bar{v}_1,\cdots,\bar{v}_{i-1}, v_i - 1\ , v_{i+1},\cdots,v_n).$$

Remark 3: If $V = V_{min}$, V^{++} cannot be defined since there exists no such i that satisfies $v_i \doteqdot \underline{v}_i$.

Proposition 3: $N(V^+) \leq N(V') \leq N(V)$ implies $V^+ \leq V' \leq V$.

Definition 4: $V \triangle V' = (v_1 \triangle v'_1,\ v_2 \triangle v'_2, \cdots, v_n \triangle v'_n)$, where "$\triangle$" means minimum.

Definition 5: $V \nabla V' = (v_1 \nabla v'_1,\ v_2 \nabla v'_2, \cdots, v_n \nabla v'_n)$, where "$\nabla$" means maximum.

Propositions 1, 2 and 3 are the same as in Sasaki et. al.[9].

3-2. Problem

Let us consider the following NLIP problem.

$$\text{Maximize} \quad f_0(X, Y) \quad \cdots\cdots\cdots\cdots\cdots\cdots (3.1)$$

subject to

$f_{i1}(X,Y)-f_{i2}(X,Y)\leq 0(i=1,2,\cdots,m)$ … (3.2)

$X-Y\leq 0$, ……………………… … (3.3)

where $X=(x_1, x_2,\cdots,x_n)$, $Y=(y_1, y_2,\cdots,y_n)$, $0=(0, 0,\cdots,0)$, and $\underline{x}_j\leq x_j\leq \bar{x}_j$, $\underline{y}_j\leq y_j\leq \bar{y}_j$, and all the $\underline{x}_j$, x_j, $\bar{x}_j$, $\underline{y}_j$, y_j, $\bar{y}_j$ are nonnegative integers; and restrict all the functions f_o, f_{i1}, f_{i2}(; $i=1, 2,\cdots,m$) to be monotone nondecreasing in each of the variables $x_1, \cdots,x_n, y_1,\cdots,y_n$.

3-3. Algorithm

In the following proposed algorithm, (3.3) is not considered explicitly as a set of constraints, but the enumeration of possible solutions is made so as to satisfy(3.3). Thus, the total number of enumerated possible solutions and the number of constraints required to test feasibility for each enumerated possible solution are smaller than those of the NLB algorithm. An optimal solution might be obtained by examining each of the possible solutions in the order of numerical ordering, beginning with($X_{max}\triangle Y_{max}$, Y_{max}) and ending with(X_{min}, $X_{min}\nabla Y_{min}$). However, this process can be considerably shortened by invoking certain rules, which are stated below.

Let (X, Y) denote the pair of vectors that is currently being examined and ($\hat{X}$, $\hat{Y}$) denote the optimal pair in the pairs of vectors that have already been examined.

Initialization. Substitute ($X_{max}\triangle Y_{max}$, Y_{max}) into both (X, Y) and (X_{max}, Y_{max}), and set $\hat{X}=\hat{Y}=0$.

Step 1. If $f_o(X, Y)\leq f_o(\hat{X}, \hat{Y})$, skip to ($X^*$, Y) and repeat step 1. Otherwise, go to step 2.

Step 2. If $f_{i1}(X^+, Y)-f_{i2}(X, Y)>0$ for some i, skip to (X^*, Y) and go to step 1. Otherwise, go to setp 3.

Step 3. If $f_{i1}(X, Y)-f_{i2}(X, Y)\leq 0$ for all i, (X, Y) is substituted for ($\hat{X}$, $\hat{Y}$) and skip to (X^*, Y). Otherwise, skip to (X^{++}, Y). In both cases, go to step 1.

In steps 1, 2, and 3, if X^* or X^{++} cannot be defined, go to step 4.

Step 4. Substitute (Y^{++}, Y^{++}) into (X, Y). If $Y\leq\hat{X}$, skip to (Y^*, Y^*), and repeat this test. Otherwise, change X_{max} to Y and go to step 1.

The algorithm terminates when Y^* or Y^{++} cannot be defined.

The flowchart and justification of the algorithm are given in Appendices A1 and A2, respectively.

4. Numerical Example

In order to illustrate the proposed algorithm, let us consider the following case:

$(c_{11}, c_{12}, c_{21}, c_{22})=(10, 10, 20, 60)$,

$(s_{11}, s_{12}, s_{21}, s_{22})=(0, 0, 6, 2)$,

$(r_1, r_2)=(0.5, 1)$,

$(C, S)=(280, 20)$.

Substituting these values into (2.1)~(2.4) yields the following NLIP problem:

Maximize $\Pi_{j=1}^{2} A_j(x_j, y_j)$ ………(4.1)

subject to

$$10x_1+10x_2+20y_1+60y_2 \le 280 \quad \cdots\cdots\cdots (4.2)$$

$$6y_1+2y_2 \le 20 \quad \cdots\cdots\cdots (4.3)$$

$$x_1 \le y_1 \quad \cdots\cdots\cdots (4.4)$$

$$x_2 \le y_2, \quad \cdots\cdots\cdots (4.5)$$

where x_1, x_2, y_1, y_2 are nonnegative integers, and (4.1) is given by equation(1). Noting that x_j, y_j(; $j=1, 2$) are integers satisfying(4.2) and (4.3), we might obtain $\bar{x}_j$, $\bar{y}_j$ as follows:

$\bar{x}_1=[280/10]=28$

$\bar{x}_2=[280/10]=28$

$\bar{y}_1=[\min\{280/20, 20/6\}]=[20/6]=3$

$\bar{y}_2=[\min\{280/60, 20/2\}]=[280/60]=4,$

where $[x]$ is the largest integer not greater than x. Thus,

$X_{max}=(\bar{x}_1, \bar{x}_2)=(28, 28)$

$Y_{max}=(\bar{y}_1, \bar{y}_2)=(3, 4).$

and the algorithm begins with $(X_{max} \triangle Y_{max}, Y_{max})=(3, 4, 3, 4)$. $X_{min}=(1, 1)$, $Y_{min}=(1, 1)$ so as not to make the steady-state availability zero.

The optimization procedure is summarized in the following table.

The algorithm for the numerical example problem terminates at 10-th iteration, and the optimal solution is $\bar{X}=(x_1=2, x_2=3)$, $\bar{Y}=(y_1=2, y_2=3)$(; x_j and y_j are the number of repair channels and total units at j-th stage, respectively), and the corresponding system steady-state availability is 0.87. Note that if the same problem is solved by NLB algorithm, 23 total iterations will be required, which is above two times than that of the proposed algorithm.

References

1. Barlow, R.E., "Repairman Problems", Chapter 2 in *Studies in Applied Probability and Management Science,* K.J. Arrow, S. Karlin, H. Scarf, Stanford University Press, 1962.
2. Barlow, R.E., and Proschan, F., *Mathematical Theory of Reliability,* John Wiley & Sons, Inc., 139-151, 1965.
3. Fox, R.J., and Zerbe, D.R., "Some Practical System Availability Calcula-

Iteration	X	Y	$\hat{X}$	$\hat{Y}$	$f_0(\hat{X}, \hat{Y})$	Relevant Steps
1	3, 4	3, 4	0, 0	0, 0	0	1, 2, 4
2	2, 4	2, 4	0, 0	0, 0	0	1, 2, 4
3	1, 4	1, 4	0, 0	0, 0	0	1, 2, 3
4	1, 3	1, 4	0, 0	0, 0	0	1, 2, 3
5	1, 2	1, 4	0, 0	0, 0	0	1, 2, 3
6	1, 1	1, 4	0, 0	0, 0	0	1, 2, 3, 4
7	3, 3	3, 3	1, 1	1, 4	0.53	1, 2, 4
8	2, 3	2, 3	1, 1	1, 4	0.53	1, 2, 3, 4
9	3, 2	3, 2	2, 3	2, 3	0.87	1, 4
10	3, 1	3, 1	2, 3	2, 3	0.87	1, 4

tions", *AIIE Transactions,* 6, 228-234, 1974.

4. Gross, D., and Harris, C.M., *Fundamentals of Queueing Theory,* John Wiley & Sons, Inc., 122-123, 1974.
5. Gross, D., and Ince, J.F., "Spares Provisioning for Repairable Items: Cyclic Queues in Light Traffic", *AIIE Transactions,* 10, 307-314, 1978.
6. Gross, D., and Ince, J.F., "The Machine Repair Problem with Heterogeneous Populations", *Operations Research,* 29, 532-549, 1981.
7. Gross, D., Miller, D.R., and Soland, R. M., "A Closed Queueing Network Model for Multi-Echelon Repairable Item Provisioning", *IIE Transactions,* 15, 344-352, 1983.
8. Lawler, E.L., and Bell, M.D., "A Method for Solving Discrete Optimization Problem", *Operations Research,* 14, 1098-1111, 1966.
9. Sasaki, M., Kaburaki, S., and Yanagi, S., "System Availability and Optimum Spare Units", *IEEE Transactions on Reliability,* R-26, 182-188, 1977.
10. Taylor, J., and Jackson, R.R.P., "Application of Birth and Death Processes to the Provisioning of Spare Machines", *Operational Research Quarterly,* 5, 95-108, 1954.
11. Wahi, P.N., "Provisioning of Spares and Service Channels when a Fixed Number of Machines Run a System", *The Journal of Industrial Engineering,* 17, 112-115, 1966.

Appendices

A1. Flowchart of the Algorithm

See the next page.

A2. Justification of the Algorithm

Justification of Step 1:

Consider any (X', Y) that satisfies

$N(X^*) < N(X') \le N(X)$

or

$N(X^+) \le N(X') \le N(X)$.

It follows, from propositions 3 and 2,

$X' \le X$

and

$f_0(X', Y) \le f_0(X, Y)$.

Thus if $f_0(X, Y) \le f_0(\hat{X}, \hat{Y})$, we need not consider (X', Y), since

$f_0(X', Y) \le f_0(X, Y) \le f_0(\hat{X}, \hat{Y})$.

Justification of Step 2:

Consider any (X', Y) that satisfies

$N(X^+) \le N(X') \le N(X)$.

It follows, from propositions 2 and 3,

$f_{11}(X', Y) \ge f_{11}(X^+, Y)$

and

$f_{12}(X', Y) \le f_{12}(X, Y)$,

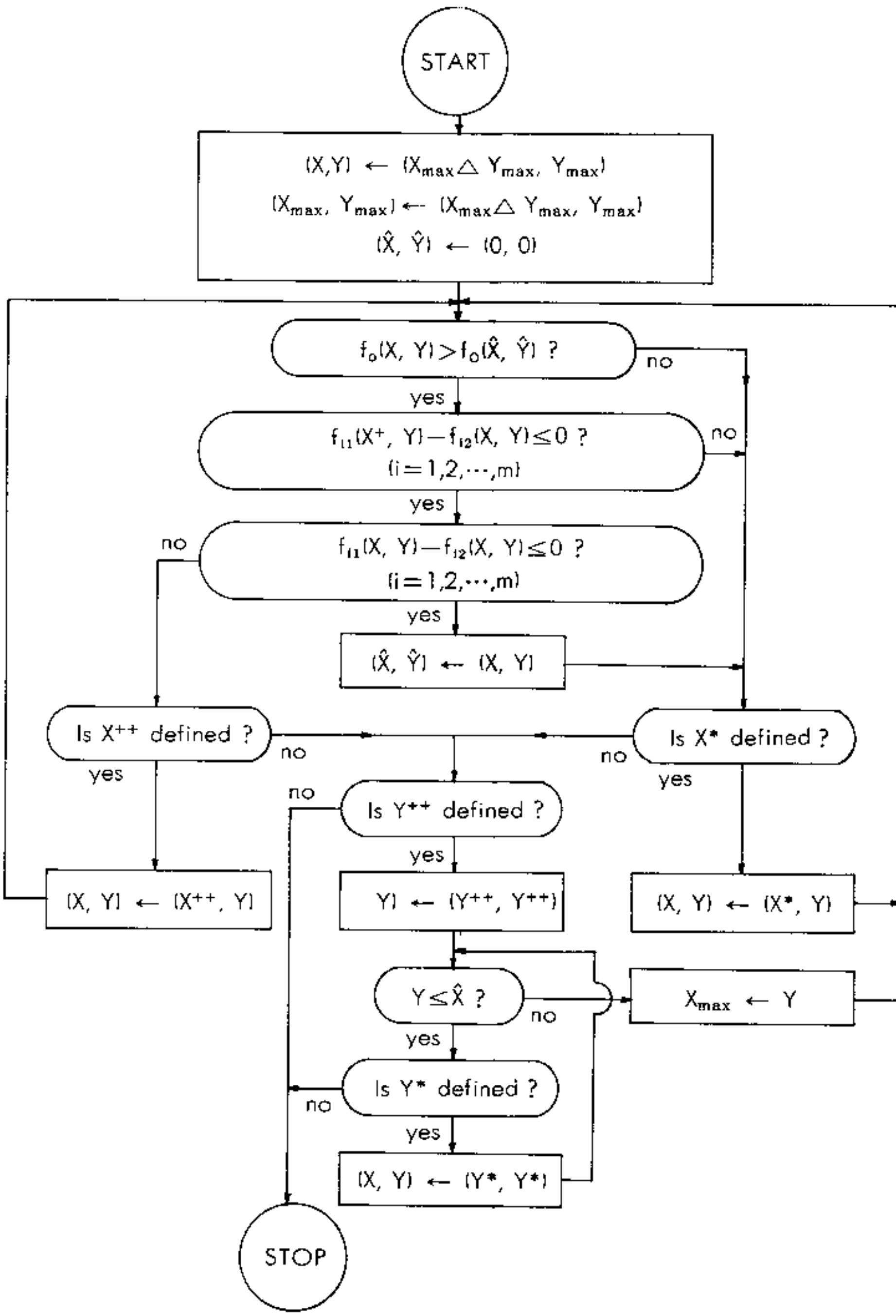
which imply

$$f_{11}(X', Y) - f_{12}(X', Y) \ge f_{11}(X^+, Y) - f_{12}(X, Y).$$

Thus if $f_{11}(X^+, Y) - f_{12}(X, Y) > 0$, we need not consider (X', Y), since

$$f_{11}(X', Y) - f_{12}(X', Y) \ge f_{11}(X^+, Y) - f_{12}(X, Y) > 0$$

and (X', Y) is not feasible.


START
(X,Y) ← (Xmax△ Ymax, Ymax)
(Xmax, Ymax) ← (Xmax△ Ymax, Ymax)
(X̂, Ŷ) ← (0, 0)
fo(X, Y) > fo(X̂, Ŷ) ?
no
yes
fi1(X+, Y)—fi2(X, Y)≤0 ?
(i=1,2,···,m)
no
yes
fi1(X, Y)—fi2(X, Y)≤0 ?
(i=1,2,···,m)
no
yes
(X̂, Ŷ) ← (X, Y)
Is X++ defined ?
no
yes
Is X* defined ?
no
yes
Is Y++ defined ?
no
yes
(X, Y) ← (X++, Y)
Y) ← (Y++, Y++)
(X, Y) ← (X*, Y)
Y≤X̂ ?
no
yes
Xmax ← Y
Is Y* defined ?
no
yes
(X, Y) ← (Y*, Y*)
STOP

Justification of Step 3:

Since (X, Y) passed step 1 without skipping to (X^*, Y), $f_0(X, Y) > f_0(\hat{X}, \hat{Y})$. Thus if (X, Y) is a feasible solution, i.e.,

$f_{i1}(X, Y) - f_{i2}(X, Y) \leq 0$ for all i,

(X, Y) is substituted $(\hat{X}, \hat{Y})$ and skip to (X^*, Y) by the same argument as in the justification of step 1. Otherwise, we should consider (X^{++}, Y), which is the first vector following (X, Y) in the numerical ordering.

Justification of Step 4:

Since (X, Y) is replaced by (Y^{++}, Y^{++}), $X = Y$. Thus $Y \leq \hat{X}$ implies $X = Y < \hat{X} \leq \hat{Y}$. Then it follows, from proposition 2 $f(X, Y) \leq f(\hat{X}, \hat{Y})$ and we can skip to (Y^*, Y^*) by the same argument as in the justificaiton of step 1. Otherwise, X_{max} is changed to Y in order to make the remaining enumeration satisfy a set of constraints (3.3) and continue the enumeration process.