A Unified Algorithm For Flowsheet Optimization

by

Y-D Lang, L. T. Biegler

EDRC-06-18-86

September 1986

A UNIFII-.D AlXOKllll.M FOK FLOWSHEET OI'TlMlZA'i'lON

V-D Lang and L. T. llegler
Depjrtmont of Chealcal Engineering
Cjmcgic-Mcl Ion University
Pittsburgh. PA 15213

Submitted to Computers and Cheaical Engineering

April. 1986

A UNIFIED ALGORITHM FOR FLOWSHEET OPTIMIZATION

YD Laity and L. T. Biefller

Clu:iuical Eiifjiiiuunng Department
CaiiMMjio-Mclloii Univoiiiiy
Pittsburgh. PA 1S213

A flexible ftlgofillmi (or flowsheet optimixaiion is developed on the FLOWTRAN piocess sinuilaiof. The opiiiniiaiion strategy combines lite feasible and uifeasible paid appioaches as well as ihe simpler black box approaches. While ihe most efficient optimization strategy is often problem dependent, this paper presents guidelines thai show which strategy is more efficient for a given problem. Also embedded within the algorithm is a new Broyden strategy for efficiently converging even complex flowsheets, without computing a new Jacobian. This allows for strategies "in-between" the infeasible and feasible path procedures. A ratio test based on the Kuhn-Tucker convergence test automatically and adaptively adjusts the optimization strategy.

The implementation on FLOWTRAN is discussed in detail and a number of examples are run to illustrate the flexibility of the implementation as welt as demonstrate the effectiveness of the adaptive optimization strategy

SCOPE

flowsheet optimization has bten an important area for chemical process design and has its origins in linear programming work in the early 60s (see Griffith and Stewart (13)1 With the increasing use of flowsheeting tools, process simulation has become easier *tui* more widespread. Currently, on commercial simulators, however, the optimization strategy is either ad hoc (i.e. often approached as a series of case studies) or involves a detached optimization algorithm that supplies sets of decision variables as new parametric cases to the process simulator. This "black box" approach was used by *Giddy* and cowoikeis [1.12J and Friedman and Pinder [11) with direct search optimization strategies, as well as by Challand [8] and Friedman and Pinder [11] with more sophisticated gradient-based optimization strategies. In these studies any gradients that were required were evaluated by perturbing the decision variables and recalculating the entire flowsheet.

Recognizing that a complete flowsheet calculation is expensive for evaluating gradients, Isaacson [15] and Parker and Hughes [20] constructed reduced quadratic models by individual model perturbation at each base point. However, these required converged flowsheets for each trial point evaluation. Moreover, recent developments in nonlinear programming algorithms have cast flowsheet optimization in a new light. Using Successive Quadratic Programming (SQP) (Han [14], Powell [22]), Berna, Locke and Westerberg [2] demonstrated with equation solving simulators that flowsheet convergence and optimization can proceed simultaneously. A number of researchers have applied this concept to sequential modular simulators [5,9,16,17] with encouraging results. While differences exist among these studies in terms of calculating gradients and implementing the optimization algorithm all of them use an infeasible path approach to optimization; i.e. tear (or recycle) streams are solved simultaneously with the optimization problem using SQP to handle both tasks.

More recently, Biegler and Hughes [6,7] advanced the concept of feasible variants, i.e. converging the flowsheet between SQP iterations. On a limited number of test problems this strategy generally required fewer SQP iterations. Reasons for this are not clear although it is easy to argue that converging the flowsheet at each iteration may help to correct problems with SQP resulting from inaccurate gradients or an inefficient line search strategy (such as the one originally proposed by Han and Powell). In more recent studies [3,4], simply improving the line search algorithm and allowing for analytic gradient information where available also improved the performance of the infeasible path approach.

Finally, Kisala [18] indicated that the feasible variant approach may not always

be superior to infeasible path. This may be especially true if either the flowsheet is difficult to converge or the SQP algorithm has little difficulty in handling the infeasible path problem. Instead, he proposed a hybrid algorithm (IPH) where the flowsheet is partially converged using a fixed number of Wegstein iterations between SQP iterations. Interestingly, this approach sometimes worked well even when compared to infeasible path. However, no criteria were given on how to choose the number of Wegstein iterations for partial convergence or how to apply this algorithm on flowsheets where the Wegstein algorithm may be inappropriate for convergence.

In this paper we develop criteria for intermediate flowsheet convergence and demonstrate this approach on a number of test problems. More importantly, however, this paper presents a unified strategy for flowsheet optimization within a fairly compact and easy to implement structure. Interestingly, this structure incorporates all approaches discussed so far and because of its implementation provides a great deal of flexibility in developing optimization strategies tailored to difficult optimization problems.

CONCLUSIONS AND SIGNIFICANCE

A flexible and efficient optimization strategy has been implemented and evaluated using the FLOWTRAN simulator. Due to the structure of the algorithms and in-line FORTRAN capabilities, the optimization implementation allows the following solution options:

- "black-box" optimization
- simultaneous convergence of recycle streams and design constraints using either Broyden or Newton methods.
- infeasible path optimization (IP)
- complete feasible variant optimization (CFV)
- partially converged flowsheet optimization with an embedded Broyden algorithm (EBOPT)

ID tins paiici we outline a swnplu donvaiiOM for *an* miprovod "black-box" sirjieijy for recycle stream problems. A detailed iloiivatiou is then piesemcd lor an cnihcdttetJ Broyden method thai parlially converges flowsheets ai intermediate uiiiiiiii.ijiiun iterations In addition, a heunstic strategy is presented that signals when pjM.ji c oiiver gcnce *s desirable 01 not. The resulting EBOPT (Embedded Oruydun Optimization) method is lairly general and leads to ihe mfeasible path and feasible vonam algorithms as limiting cases.

The EBOPT strategy is compared to Ihe infeasible path hybrid algorithm (IPH) developed by K.sala and some theoretical advantages of EBOPT are demonstrated. In particular. EBOPT can generally converge more complex flowsheeting problems more efficiently because of its Broyden capabilities. Also. EBOPT is not as prone to line sea»ch failures as IPH is.

Fmaiiy. the capabilities of the optimization implementation are demonstrated on five e*ampic problems. The first problem is characteristic of black-box optimization while trie second illustrates the flowsheet convergence capabilities afforded by the EBOPT and IP algorithms.

The last three examples give a comparison of four algorithms. IP, EBOPT. CFV and IPH. on reasonably difficult and realistic flowsheet optimization problems. On all problems EBOPT performs better than either IP or CFV. IPH performs best on one problem but suffers premature line search failures on the other two.

1. Preliminary Theory and Concepts

The flowsheet optimization problem is given by

(NIP) Mm F(x.y)

SI. h(x.y) • y - w(x.y) • 0
cL.y) • 0
gU.y) * 0
K S' * £ *

$y_l \le y \le y_u$

where
x - flowsheet decision vanahios
y - guossed lea' stream variables
w - calculated teai stream vanatjius
F - oi>|cciive function
h - ie«< equations for converging the flowsheet
c - additional equality constraints lor optimisation
g - inequality constraints

Examples of objective and constraint functions can be found in previous studies as well as in Ihe case studies presented later *in* the paper. To solve this problem, the Successive Quadratic Programming algorithm essentialy applies a modified quasi-Newton method to converge Ihe optimally or Karush-Kuhn-Tucker (KKT) conditions of (MLP). To do this and maintain a consistent active set. the following quadratic program (QP) is solved at each iteration:

(QPl)

$$\min \ \nabla F(x^i, y^i)^T d + 1/2\, d^T B d$$

$$\text{s.t.} \quad h(x^i, y^i) + \nabla h(x^i, y^i)^T d = 0$$

$$c(x^i, y^i) + \nabla c(x^i, y^i)^T d = 0$$

$$g(x^i, y^i) + \nabla g(x^i, y^i)^T d \le 0$$

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} \le \begin{bmatrix} x^i \\ y^i \end{bmatrix} + d \le \begin{bmatrix} x_u \\ y_u \end{bmatrix}$$

Here B Is a BFGS (see (10]) update matrix to the Hessian of the Lagrange function with respect to x *»nd* y.'A detailed statement of this algorithm may be found in any of the above references end will not be given here. The version of the algorithm used in this implementation was developed in Biegler and Cuthrell [4] and includes the following features:

1. All eHicioMi augmented Lag*ang.an-based line search strategy is used lo guarantee ylobal convergence and allow lull slops in lha region of tha opinnum. The latter ptopariy is not guaranteed by ll»e implementations of *Hsn* (14) 01 Powell (22].

2. An automatic variable and constraint scaling strategy is included that gives good performance *on* flowsheeting problems. In addition a condition number is calculated for lha 8 matrix in QP1 to del ermine when tha problem is ill-conditioned or poorly scaled.

3. BecA^c of portability, space and availability constraints, tha Ha*we11 subiouiine VE02AD is used to *o4ve lha OP at each iteration. In Biegier and Cuirvell (4) a more reliable and efficient OP coda was used. However, there *f* no noticable differences in function evaluations due to this substitution and. consequently, tha FLOWTRAN implementatlon is not affected by this change.

This SOP algorithm forms lha core of our unified optimization strategy.

To see how the optimization strategies compare from a geometric viawpoint. consider the sample flowsheet optimization problem in Figure la. Hera only one decision variable. *. one tear variable, y. and only one taar aquation, h. are required for the optimization problem. If one considers this problem from a casa study perspective, one can trace a curve for Fix) vs. x (Fig. 1b) where each point on tha curve represents a converged flowsheet. Expanding this problem *in* terms of both x and y yeids the contour plot in Fig. 1c. Note that tha optimum lias on the solid ime which represents the tear constraint and a nonlinear projection along this line gives the curve m Fig. 1b.

Us.ng the case-study or "black box" approach tha optimization algorithm is merely tied to the outside of the simulator and tha simulator Is responsible for converging the flowsheet for each evaluation of tha optimization problem. Similarly, gradient calculations involve perturbation *snd* convergence of the flowsheet for each decision variable. Thus, no information about flowsheet convergence is passed between the optimizer and simulator. In Figure 2 this can be seen in terms of tha horizontal steps (in x) made by tha optimizer and tha vertical steps (in y) performed by the simulator. Note that these vertical steps usually represent flowsheet

cunvOMjunce by slowly converging lecyde alyu* iiitinj mJ ih«iefore io|)iesa<H Hie most lime consuming pen of Hie opi»»niieuon duOr

Since the mfeasible path approach consiiucit information about tha »-y suHace and does not require flowsheet convergence until the optimum is found, movement occurs in both x and y as seen *in* Fig. 3a. This slap in x and y results from linearizing lha constraints and approximating the surface contours as wall tha curvature of tha constraints. As saan above this approximation laads to a straightforward quadratic program. Gradients for (OP1) *im* found by perturbing lha unconverged flowsheet Also th« expansive vertical staps for flowsheet convergence are avoided because flowsheet convergence is guaranteed as part of tha solution to tha optimization problem. In fact, as will be illustrated later, application of tha infaasibla path approach in tha absence of degrees of fraadom is equivalent to Newton's method.

To prevent an ovaraxtrapolation of tha infaasibla • path approach it may ba advantageous to ensure that tha equality constraints ba converged (or at laast partially converged) at each Iteration. Using tha faasibla variant approach, tha path for x and y Is given by Figure 3b. *Htm* vertical staps from flowsheet convergence are introduced and one seas that lha starting point for converging My) • 0 is given by tha OP and is considerably belter than in tha "black-box* approach. Interestingly, the OP that Is created and solved at each Iteration Is axactly tha same, and requires tha same effort at each Iteration, as with infaasibla path. Nota that with this strategy one assumes that tha flowsheet can ba solved raadily by tha racycla convergence algorithm.

In lha next sections we develop a new approach for Improving tha performance of the optimization strategy. This stratagy addresses soma of tha drawbacks with both infeasible path optimization and tha faasibla variant stratagy and also links both strategies more closely in terms of a unified framework. Before presenting this strategy, however, it is useful to discuss further the differences between feasible variants and "black box" (or case-study) optimization.

2. Black Box Optimization vs. Optimization in x and y

Comparing Figure 2 to Figures 3a and 3b one sees that the main disadvantage in the optimization path of the first figure is due to lack of interaction between x and the dependent variables, y, in the optimization step. In fact, the main difference between Figures 2 and 3b is simply that with the feasible variant approach y is initialized much closer to the converged flowsheet. Generally, this leads to more efficient recycle convergence. The improved path however requires flowsheet perturbations in x and y to create a larger QP problem at each iteration. Using SQP with the "black box" approach requires the solution of a much smaller QP:

(QP2)
$$\begin{aligned} \text{Min} \quad & \frac{dF^i}{dx}^T d_x + \frac{1}{2} d_x^T B_x d_x \\ \text{s.t.} \quad & g(x^i) + \frac{dg^i}{dx}^T d_x \le 0 \\ & C(x^i) + \frac{dC^{iT}}{dx} d_x = 0 \\ \text{where} \quad & y = y(x^i) \,|\, h(x^i, y) = 0 \end{aligned}$$

with the tear equations and tear variables removed. Note that the derivatives in the above QP are reduced gradients and require a converged flowsheet with each decision variable perturbation.

Because of the differences in the size of the optimization problem it is easy to see that the "black box" approach can be superior to the infeasible path or feasible variant methods *when* the number of tear varibles greatly outnumbers the number of design variables *and* the flowsheet is not difficult to converge with conventional algorithms. The work per iteration for each approach can be approximated by:

Black Box

$$NFPI = NRP \cdot NX + NRC$$

Feasible Variant

$$NFPI = E \cdot NX + NY + NRI$$

Infeasible Path

$$NFPI = E \cdot NX + NY + 1$$

where

	NFPI	number of flowsheet passes per iteration
	NRP	number of recycle iterations to converge perturbations in decision variables
	NRC	number of recycle iterations to converge flowsheet at each new base point (vertical steps in Fig. 2)
	NX	number of flowsheet decision variables
	NY	number of flowsheet tear variables
	E	fraction of equivalent flowsheet passes required for decision variable perturbations (partial flowsheet passes)
	NRI	number of recycle iterations to converge flowsheet at new base point (vertical steps in Fig. 3b)

As seen from the above relationships, flowsheets with few degrees of freedom and many recycle components can be optimized more efficiently with the black box approach. Note also that NRI is expected to be less than NRC. This occurs because the y variables have better initialization with the feasible variant approach and, as will be seen later, more efficient recycle convergence algorithms can be used with feasible variants. With the black box approach it is easy, however, to reduce NRC and allow the optimization path to be similar to the one followed by the feasible variant approach. This simply requires keeping track of how the dependent variables, y, change with x.

Consider a perturbation in variable x and a completely converged flowsheet for that perturbation. For the tear constraints, $h(x,y) = 0$, we have to a first order approximation:

dh • 0 - V *h\im* • V *h'dy*

where dx • (0. 0.....Ax.....0J. Solving this aquation foi dy/A*(give* column | oi Hie matrix -<V Mx\ y')[1]) 'V Mx*. y'l*. Therefore, simply by saving the response of ihu y variables to all perturbations *in* x. i.a.

X - *tdy%/A*t.dyj/&Mt. . . . .<*Ym%/**mw)*

- -(V />UV)VV /KJ»'.KV •

we can use the solution of (QP2). d^. and wrila d_y as

d • *Y d* .

Note from QPi that this equation solves the linaariiation of the tear constraints and therefore leads to the step in x and y givan In Figure 3b. Also, it is interesting to note this approach leads to the same slap thai is generated by the Reduced Feasible Variant (RFV) algorithm described by Biaglar and Hughes (7).

However, because flowsheet convergence Is the outar loop to saveral levels of iterative calculations, the convergence error In the tear aquations can ba relatively large. Therefore, in order to calculate the Y matrix correctly, lha perturbation size needs to be chosen accurately. If we include second order corrections and the convergence error. {. in our tear constraints we can write:

$$dh - i - VA'AJT \cdot (HA^{**}) \cdot \nabla_y h^T \Delta y + O(\Delta x \|\Delta y\|) + O(\|\Delta y\|^2)$$

Rearranging this expression gives an ordar of magnitude estlmata for tha errors in the Y matrix:

$$(Y)_j = (\nabla_y h^T)^{-1}\left[\epsilon/\Delta x_j - (\nabla_x h^T)_j + O(\Delta x_j) \cdot (X|AK|) + O(\|\Delta y\|^2/\Delta x_j)\right]$$

Note that choosing a perturbation alxa too small laada to appreciable error due to convergence noise while a large perturbation alza leads to an error due to second

order cllucit. To avoid these problems. Uso number of Heranonv NHP and NflC. may nood io tse large IO force a amsii mo' Tiwt it •tp«ci«Hy tiu« boctuit flow*)**** convex (jcrxe elgoriihme have only *unmmt* c unve^u+nce p*op«m«i HHJl lequtrtog MPi io be larger ihan it normally e*peci»<J *tw* •imulattofv

3. Development of an Embedded Broyden Strategy

To summarize tha previous material and to introduce this section, consider problem (NLP) again:

(NLP) Min FU.y)

s.t. Mx.y) • y - w(x.y) • 0
c(x.y) • 0
gU.y) 1 0

$$y_l \le y \le y_u$$

In tha "black box" approach tha y variables ware eliminated and tha constraints. hU.y) • 0. were always satIsflad. *rnvn* for parturbatlons of x. Tha infeasible path (IP) and tha completa faaslbla variant (CFV) atrataglaa daal with iNLP) explicitly In tha space of x and y and aolva (QPI) at each Iteration. In addition. CFV converges tha equations. h(x.y) • 0. by adjusting tha y variables *iftir* (QPI) Is solved. As mentioned above, it may ba more efficient to *psri/sHy* converge tha constraints. htx.y) • 0. at each iteration since IP yields a converged flowsheet at the optimum anyway. Also. If an afficiant and rellabla aquatlon-«olv'€r can ba applied, one can handia both h and c by convarging tham almuJtaneously.

In this section. we present en ambaddad Broydan approach for partial convergence within flowsheet optimization. This strategy Incorporates tha Infeasible path and faaslbla variant approaches as limiting casaa and can be vlawad as a modification of lha hybrid approach proposed by Kisala (18]. In tha previous section we observe that slowly convarging recycle algorithms can laad to inefficiency wiih the black bo* optimization algorithm. These algorithms log. WiMjbiom o» direct substitution) ere also used in tlie feasible variant and hybrid apfwoaches. Thus, for flowsheet thai aie difficult lo lolve. one can e«i>oci conveigcuce problems ai intermediate points.

When tear variables and constraints ere perl of the oplimizeiion problem, we have enough gradient information from QP1 to allow also for more efficient convergence routines. In particular. Broyden routines have been used with good results (9. 19. 21] m converging complex flowsheets, even those with additional uesign constraints. This section therefore add/asses two points.

1. HOW can a Broyden algorithm be embedded within the optimization strategy to (partially) converge the flowsheet at intermediate points?
2. What criterion should be used to decide whether (partial) convergence is necessary at intermediate points?

3.1 An embedded Broyden algorithm

we now derive a modified Broyden algorithm for partial convergence et intermediate points. For convenience we will use only the tear variables, y. and leer equations. nU.y) > 0. in presenting this derivation. Application of this method to additional design constraints, cU.y) • 0. end additional dependent variables is straightforward.

To converge or partially converge the equality constraints, consider the step in Figure 4. At point C. the gradients end values of the objective and constraint functions ere evaluated and QP1 is constructed end solved. The search direction from QP1 and a suitable stepsize leads to point D. from which the equality (I.e. tear and any design) constraints may be converged. If one were to apply a Broyden method to converge the dependent variables et this point one would want to have the Jacob.an (V^h)' at point O to initialize the Broyden method. Since this is not available and it would be expensive to construct this information, we derive en update strategy based on the gradients evaluated at point C.

Lot H" • $[\nabla_x h^1 \; \nabla_y h^1]$ at iiouii C and consuloi tlie Broyden formula (10J

(Hi) $$H^{k+1} = H^k + (q - H^k \delta)\delta^T / \delta^T \delta$$

$$\text{where} \quad \delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} - \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

, - he,"*VS - h<.V>

We note thai this update relation can also be epplied to the *nonsqumf* metrix without violating any assumptions (see Dennis & More [10]) es to its derivation. Also from QP1 we see thel the slep from C to D is genereted by H*d • -t»(x^c. y^c). We cen now apply the updete formula to get H^1 at point 0;

<B2) $$H^1 = \left[\nabla_x h^T + \frac{(q - H^0 \delta)\, \delta_x^T}{\delta^T \delta} \;\vdots\; \nabla_y h^T + \frac{(q - H^0 \delta)\, \delta_y^T}{\delta^T \delta}\right] = \left[H_x^1 \;\vdots\; H_y^1\right]$$

Starting from point O we keep x constant and only change y to converge the flowsheet; thus δ_x-0 for the following Iterations. Applying (B1) to H' gives the following relations:

(B3) $$H^{k+1} = \left[H_x^1 \;\vdots\; H_y^k + \frac{(q - H_y^k \delta_y)\, \delta_y^T}{\delta_y^T \delta_y}\right]$$

and

<B4) $$\left[H_x^1 \;\vdots\; H_y^k\right] \begin{bmatrix} 0 \\ \delta_y \end{bmatrix} = -h(x^D, y^k)$$

Note that since J_ is determined by QP1 for the first step t\d H_a does not affect δ_y in the later steps, we write the updete formulae (B2). (83) es:

$$H_y^1 = H_y^0 + (q - H_y^0 s_y)\, s_y^T / s_y^T s_y$$

and

$$H_y^{k+1} = H_y^k + (q - H_y^k s_y)\, s_y^T / s_y^T s_y \quad for \quad k \ge 1$$

This expression can be simplified *mvmn* further by noting that

$$H_y^k s_y = -h(x^0, y^k)$$

and

$$q - H_y^k s_y = \left[h(x^0, y^{k+1}) - h(x^0, y^k) \right] + h(x^0, y^k)$$

if /u// Broyden steps are taken. Also for the first step:

$$H^0 s = H^0(\lambda d) = -\lambda h(x^c, y^c)$$

Therefore, the Broyden steps can ba given by:

$$(8b) \qquad H_y^1 = H_y^0 + \left(h(x^0, y^1) + (1-\lambda) h(x^c, y^c) \right) s_y^T / s_y^T s_y$$

and

$$(B6) \qquad H_y^{k+1} = H_y^k + h(x^0, y^{k+1})\, s_y^T / s_y^T s_y\,, \quad k \ge 1$$

Note that when the line search in SOP allows a full step U-I). (B5) and (B6) differ only in the denominator of the second term. Also, note that In Iha absanca of degrees of freedom (no x variables), thasa aquations reduce to the conventional Broyden approach.

Testing of this approach on the problems given below shows that no more than S iterations are required to converge the flowsheet at intermediate points. As shown

with the conventional Broyden approach. Hits method also handiei t)e»ign conairamit easily.

3.2 Criteria for using en e«nb«Sde>d B#oyd«n method

As noted in previous studies, the choice of optimization strategy is often problem dapandant. If Iha gradients are reasonably accurate and iha flowsheet is only "mildly" nonlinear, than iha infeasible path approach will converge smoothly. If. on the other hand, iha flowsheet Is highly nonlinear and difficult lo converge (with complex units thai are failure prone) then a faasibla path approach with Broydan's method and appropriate safeguards could ba more reliable and efficient. However, these characteristics *mm not* always known *a priori*. Indeed, *M* shown by Kisala [18]. partial flowsheet convergence may lead to more efficient performance In solving the optimization problem.

However, care must ba taken In dealing with partial flowsheet convergence at each iteration. In particular, partial convergence can ba detrimental to the Una search algorithm in determining the stepslie for the next point.

In the SOP algorithm, a given stapsiza along the search direction, d. is accepted if a "sufficient" decrease is observed with some merit function, p. In the algorithms of Han and Powell, this function is the exact penalty function; in our algorithm an augmented La grange function is used. In either casa. it is wall known (sea e.g. Han 114)) that tha search direction from QP1. d. is a descent direction for the merit function. Consequently, finding a nonzero stapsiza Is guaranteed, at least in theory, for tha infeasible path algorithm.

For tha feaslbla variant algorithms, one can also prove that a nonzero stapsiza will ba found during tha lina search. This can ba shown because all points In tha line search hava converged equality constraints and tha OP solution, from a faasibla point, is also a descant direction for tha lina search function *f*.

However, *lot* flowsheets thai ate only partially converged, ont *csntwl* guarantee iliat a stepsue with a decreased merit function will be found. A simple illustration of this is given in Figure S. From the QP base point at A. one sees the mem function can be viewed as a function of 1 along the search direction found by OP 1 Executing a fi*ed number of convergence iterations for a given siepsiie. point B. say. *m*y* decrease or increase the objective function. In fact, for a fixed number of iterations, it is possible that the equality and inequality constraint infeasibilities may also increase. Consequently, it is possible that partial convergence may move the mem function value from point 8 to B*. Note that this behavior can occur arbitrarily close to point A. say. point C. and thus lead to a line search failure - even though perfectly reasonable stepsizes exist for infeasible path.

For this reason we apply partial convergence only *sfter* the line search algorithm finds a stepsize. In this way we can avoid line search failures and also save some work at intermediate points.

As further justification for this safeguard we note that, by itself, the infeasible path algorithm converges quickly and takes full steps in the neighborhood of the optimum. For this case, partial convergence is usually not necessary. On the other hand, at the beginning of the optimization, the search direction may overextrapolate and lead to a point that is difficult to converge. Here it would be Inefficient to partially converge the flowsheet during the line search. Instead, allowing the line search algorithm to find a more reasonable point first will save some effort.

Even with this safeguard, one is still not guaranteed that partial convergence leads to better performance for the optimization. One way to measure the success of partially converged points would be to compare merit functions from iteration to iteration. However, one still needs to know if. a *priori,* partial convergence is desirable or even necessary at a given point. In the next section we develop a strategy for dealing with this task. We should mention that this strategy is based on

heuristics and. consequently, will not always guarantee improved fw formeitce competed to inf«a»ible path No*ivn K,« •••uh» ihai ~« f><o»«« I«I«« *• *•• r encouraging

Heuristic strategy for partial convergence

For SOP a common measure of Kuhn Tucker error (KTE) for problem INLP) is given by (22):

$$KTE - |\nabla f\backslash/| \; \sum_i K^*,.I \bullet \sum_i I^{**} I \bullet \sum_i I'<,I$$

where u_i, v_i and $l^\wedge$ are multipliers calculated for QP1 for g. h_i and c_i respectively. Also. $9_i(U.y)$. is defined as max (0. g^x.yM. Reducing KTE to within a zero tolerance is a necessary and sufficient condition for satisfying the KKT conditions for (NLPL Now. from Figure 4. if SOP finds a step from point C to point 0. one needs to determine if additional work is required by Broyden's method to move to point E. Since this method converges only c_i and h_i a *heuristic* measure of how much improvement can be had is given by the ratio:

$$\text{(BR)} \quad \left(\sum_i |v_i h_i| + \sum_i |t_i c_i|\right) / KTE$$

If this ratio remains small, intermediate convergence is not necessary. If it remains consistently largo, however, full or partial convergence may help to speed convergence. To us« this ratio we propose two triggers for intermediate convergence.

$$« \quad " \left(\sum_i |"A| \bullet \sum_i |v,|\right)_{\bullet} / KTE_{\bullet} > \epsilon_1$$

at point C. mov« to point D *mnd* converge c_i and ^ (to point E, say) until the relation

$$\text{(BR1)} \quad \left(\sum_i |v_i h_i| + \sum_i |t_i c_i|\right)_{\bullet} / KTE_{\bullet} \le \epsilon_2$$

is satisfied.

2) *if* $\left(\sum |v_i h_i| + \sum |t_i c_i|\right)_C / KTE_c \leq \varphi_1$

but on moving to point D,

$$\left(\sum |v_i h_i| + \sum |t_i c_i|\right)_D / KTE_c > \varphi_2$$

use Broyden's method to converge (to point E, say) c_i and h_i so that (BR1) is satisfied.

Otherwise, both points C and D lie close to the constraints and intermediate convergence is not required. Note that by adjusting the φ's, one can develop a full spectrum of methods between the infeasible path approach (φ_1=1, φ_2=∞) and the feasible variant strategy ($\varphi_1 \approx 0$, $\varphi_2 \approx 0$).

In choosing these parameters, φ_1 should be set between zero and one to allow for partial convergence. φ_2 should be set small in order to avoid the first trigger at the *next* iteration. Our experience indicates that often very few Broyden iterations (1 or 2) are required to satisfy (BR1) even if φ_2 is small, (say 10^{-3}). φ_2, on the other hand, can be sufficiently greater than unity without hampering performance. This results because point C for the second trigger is sufficiently close to satisfying the constraints; a linearization from that point and a line search usually determine point D that is reasonably good without partial convergence. In fact, for the problems we solved, the second trigger for partial convergence was not necessary for good performance.

In addition to the above triggers we have also included the following conditions for intermediate convergence. First, if the current iteration is in the neighborhood of the optimum, applying Broyden iterations is usually not necessary since the constraints are close to being satisfied anyway. Therefore if $KTE_i \leq 10\epsilon$, say, where ϵ is the Kuhn-Tucker tolerance, we do not apply intermediate convergence.

Also, it is possible that the flowsheet may not converge at all at an intermediate point. Consequently, we impose a maximum number of iterations for intermediate convergence. In our case studies successfully converged intermediate points never required more than 5 iterations.

We conclude this section by emphasizing that the above strategy is based on heuristics that, from our limited experience, work reliably and efficiently. Since very little theory governs the concept of partial convergence, we used these guidelines in our implementation. In the next section we discuss how the constants φ_1 and φ_2 were chosen and give a statement of the algorithm.

4. Algorithmic Statement and FLOWTRAN Implementation

Using the concepts stated above we now present an algorithmic statement of the Embedded Broyden Optimization (EBOPT) strategy and outline the features and options used in the FLOWTRAN implementation. In the algorithmic statement we assume the reader is somewhat familiar with the SQP algorithm and will not dwell on its details. The reader is referred to [4] for the line search strategy and update formulae.

4.1 Algorithm

Step 0) Set the SQP iteration counter, i=0, and initialize the flowsheet with x^o and y^o. y^o can be found by (partially) converging the flowsheet. Set ϵ as the Kuhn-Tucker tolerance.

Step 1) At (x^i, y^i) find the gradients for F, g, h and c with respect to x and y. This can be done by direct loop perturbation [6] or chainruling [3].

Step 2) Solve (QP1) given above to get the search direction d for x and y. Evaluate KTE and (BR) at iteration i using the expressions above. If $KTE_i \leq \epsilon$, stop.

Step 3) Perform a line search with a suitable merit function, f, to find a stepsize, 1, along d. Define ^"•«'«kdᵢ. ỹ^y^Xd^.

Step 4) Set Broyden iteration counter k«0 and evaluate the flowsheet (or <*'". ỹ).

If $\left(\sum_i |v_i h_i(x^{i+1},\tilde{y})| + \sum_i |t_i c_i(x^{i+1},\tilde{y})|\right) / KTE_i \le \epsilon_1$

or

KJE_i i 10«

set y^{i+1} • ỹ\ go to step 7.

Step 5) Set y* • ỹ* and apply the Broyden formulae (B5) and (B6) to y^k and (h.c) until:

$$\sum_i |v_i h_i(x^{i+1},y^k)| + \sum_i |t_i c_i(x^{i+1},y^k)| / KTE_i \le \epsilon_2$$

then set y^{i+1} * y^k. If the above relation cannot be satisfied after five iterations (w>5). set y" • y\

Step 6) Evaluate the gradients at (x^{i+1}, y^{i+1}) as in step 1*. Update the Hessian matrix for QP1.

Step 7) Let i»i*1 and go to step 2.

4.2 FLOWTRAN Implementation

The optimization capability in FLOWTRAN was Installed by writing a type 2 (convergence) block. The structure and argument list for this block, called SCOPT. was the same as the existing recycle convergence block, SCVW. Because we did not change any code in FLOWTRAN. we Implemented direct loop perturbation as the most straightforward way for evaluating gradients. Since FLOWTRAN generates end compiles a FORTRAN main program at run time, it can easily accommodate in-line FORTRAN and user written subroutines as part of the input data. This, In turn, allows the optimizer to evaluate partial flowsheet passes if needed for gradient evaluation.

Aim. ih« user • specification ol the opiunuanon piobiem can be nude umply by aOUiiMj a few lm«t ol «n line FORTRAN to the input daia lor the iiiiiniiiiun problem

SCOPT handles up to ilwee tear streams (as does SCVW). which it converges simultaneously, and up to a total of 40 decision and leer variables. Decision variables can be chosen from equipment parameters or feed streams. These variables are accessed through PUT statements that ere common features in FLOWTRAN.

In addition, the user needs to specify a reletive perturbation size (between 10' and 10' is recommended) and a reletive Kuhn-Tucker tolerance (usually between 10*' and 10*). It should be noted, however, that choosing a Kuhn-Tucker tolerance too small can result in line search failures and poor steps near the end of the run. because the gradients mey not be accurate enough to satisfy the tolerance.

Another option In our implementation deels with the choice of tear variables. In most optimization studies, stream flow rates, pressure and specific enthalpy are chosen. Temperature Is not chosen because for multiphase streams, enthalpy is not realizable from temperature alone. However, for *s/ng/t* phase streems calculation of enthalpy from temperature is usually direct *Bnd* a level of iteration and some convergence noise ere eliminated in the perturbation step. Since perturbing the temperature for single phase tear streems can lead to more accurate derivatives, we have added a T/H option.

Finally the t parameters need to be specified for the embedded Broyden algorithm. As mentioned above, f_y which defines the second trigger, can be fairly large. In our experience values of f_j > 3 have still led to good performance end this trigger was always Inactive. Consequently, we have not used this test et ell.

On the other hand ϵ_f was set. after some testing, to 0.01. As explained above, it takes surprisingly few Broyden iterations to satisfy this test. The most Important parameter for determining intermediate convergence is therefore ϕ_1. Setting $\phi_1=\phi_2\approx 0$ in our implementation leads to a feasible variant approach. Setting $\phi_1=1.0$ yields the infeasible path approach and intermediate values of ϕ_1 allow partial convergence. In our testing, setting ϕ_1 to 0.4 yielded good results although this parameter is problem dependent. However, on many problems, examination of the output shows that performance of this strategy is not very sensitive to ϕ_1. In Table 1, the ranges of ϕ_1 and ϕ_2 under which the *same* performance would be achieved are tabulated for the Embedded Broyden Optimization (EBOPT) strategy.

5. Example Problems

The following five example problems were solved by a number of approaches. A number of similar problems were solved in addition to these. However, for the sake of brevity, we chose this set because it represents what can be expected from the implementation in terms of performance and flexibility. The first problem is essentially a black-box implementation on a single unit. The second problem illustrates how the infeasible path and Broyden methods can be used to converge flowsheets. The last three problems deal with moderately-sized flowsheets, some with complex models. These allowed comparison of the embedded Broyden strategy (EBOPT) with a number of recent and efficient optimization strategies. Due to the flexibility of the algorithm and features of FLOWTRAN, none of these strategies was difficult to implement.

Problem 1 - Black Box Optimization

The first problem deals with a single unit optimization of a 25 tray distillation column with sidestreams. As illustrated in Figure 6, the distillation column problem, which is solved by a Thiele-Geddes model (FRAKB), seeks to maximize the degree of separation of its 5 components among its overhead, bottoms and sidestreams. The decision variables are the fraction of feed to the two sidestreams and the distillate.

The only constraints are bounds on the decision variables as well as bounds on the fraction of feed to the bottoms stream.

This problem is typical of many simple process optimization problems. Since there are no recycles, SQP deals with this model in "black-box" fashion and solves it completely each time it requires a function evaluation. Alternately, an entire flowsheet could easily have been treated instead of a single unit. The solution of this problem is also given in Figure 6. This problem was solved to a relative Kuhn-Tucker tolerance of 10^{-6}.

Note from Table 1 that the performance of the optimization algorithm is characteristic of the black-box approach. Because the model needs to be solved several times it is not surprising that over 16 Simulation Time Equivalents (STE's measured at the starting point) were required to optimize this three variable problem. Because of the tight tolerance, seven iterations appears to be reasonable for this case.

Problem 2 - Cavett Problem Simulation

To demonstrate the capability of the infeasible path and EBOPT methods for Newton and Broyden convergence, respectively, we selected a modified form of the Cavett problem, reported by Rosen and Pauls [23]. Here the number of stream components was reduced from 16 to 11. Figure 7 illustrates the flowsheet where Z1 and Z2 were chosen as tears and Table 2 lists problem data and the converged solution. This problem was first solved using the Wegstein convergence block in FLOWTRAN with all of the default options. In this case 13 iterations were required to converge the flowsheet to the default relative tolerance of 0.0005. Using the same tolerance, this flowsheet was also converged using the infeasible path (IP) and EBOPT methods in 4 and 6 iterations, respectively. However, comparing STE's for this problem shows that these methods are not competitive with Wegstein. The

EBOPT method retinues 26 flowsheet passes lo construct a Jacobian matrix ai ihe fusi iteration, ihe IP incihod oouils 10 constiucl Ihis Jacobian ai every iteration.

Because of ihe effort required for the initial Jacobian. Broyden's method may not always be competitive for solving simulation problems. Embedded within an optimization strategy, however, where the Jacobian Is calculated anyway. Ihe Broyden method performs much more efficiently.

For the next three examples we compart the IP and EBOPT strategies with the CFV (Complete Feasible Variant) [7] and IPH Unfeasible Path Hybrid) [18] algorithms. The last two algorithms were implemented by using FLOWTRAN's Wegstein convergence block to (partially) converge the flowsheet between SOP iterations. For IPH. two wegstem iterations were used between every SOP iteration, as suggested by Kisala (18], For *CFV.* the flowsheet was either converged to FLOWTRAN's default tolerance or until 30 Wegstem iterations had been exceeded.

On all problems relative tolerances of 10^{-4} were used for the Kuhn-Tucker error. All problems were recycle flowsheets with complex unit operations and nonideal thermodynamics.

Problem 3 - Ammonia Process A

This problem was adapted from Parker and Hughes (20] *»nd* has been used in other studies 19. 18]. The problem statement is given in Figure 8 and in (20). Because of different thermodynamic properties and fewer decision variables, values of the objective function are slightly lower in this study. The starting point and optimal solution for this problem *»f* given In Table 3. As seen from Table 1. the double loop flowsheet with *»n* equilibrium-based reector Is fairly easy to converge and optimize. Here the EBOPT and CFV approaches are close in performance. although EBOPT is slightly superior. Because no Intermediate convergence was applied for the IP run. more iterations were required than with EBOPT. Interestingly.

EBOPT. with the heuristic strategy detcnUed in Section 3. only n«·c«iu<f to use Hi« Bioyden strategy after iterations 1 and 3. After iiicsc. SOP r**d no irouMe converging tins problem by itself.

Unfortunately for this problem, the IPH algorithm suflered a line search failure after 5 iterations. Restarting at this point resulted in a second line search failure after 3 additional iterations. In his study. Kisala (18) also reported a line search failure for Parker's ammonia problem. The reason for this, as explained in Section 3. may be that, because a fixed number of Wegstein iterations *Mt9* applied for each function evaluation in the line search, a descent direction cannot be guaranteed and this method can be prone to failure.

Problem 4 - Methytchlorobenzene Process

This problem is adapted from *an* example in the FLOWTRAN manual (24). Using the default costs and prices In the costing blocks, the optimization problem illustrated in Figure 9 was formulated. Here six decision variables were chosen for the optimization. These ere listed along with their initial and optimal values In Table 4.

Because this problem contained a rigorous (and often unreliable) absorber model and the FORTRAN code for FLOWTRAN was not available to us. we were unable to provide error returns to Ihe optimization algorithm and thus continue in the event of unit convergence failures. Obviously, error returns are a necessary feature In the Implementation of any flowsheet optimization strategy, and the lack of this capability reflected how we could solve this problem.

From the results In Table 1. one sees that the EBOPT strategy required less effort than either the CFV Of the IP strategies. However, to prevent premature termination due to failure In the absorber block. Intermediate recycle convergence was suppressed for EBOPT during the first two SQP iterations. The EBOPT algorithm nee.led to apply Broyden's method only afiar iteration 4 in oidar to gal aalitfaciory performance.

Also, duo 10 difficult!** with lha abtorbar. lha IP algorithm could not be converged from the starling point for EBOPT. From a slightly diffarant starting point (shown m Table 4). 12 iterations were required to satisfy a Kuhn-Tucker tolerance slightly above 10'*. Again, because of the unrellable nature of the process units, a belter and more consistent comparison could not be) made.

The CFV algorithm required over S times the computational effort that EBOPT required. This represents the difficulty that SCVW has to converge this flowsheet at intermediate points. In fact, for SOP Iterations 1, 2. 5 and 0. CFV required the maximum of 30 iterations without converging the flowsheet at these base points.

Again, as with the previous problem. IPH terminated with a line starch failure after 12 iterations. This could be due to the descent direction line search problem explained in Section 3.

Problem 5 - Ammonia Process B

The flowsheet for this problem Is given In Figure 10 along with the problem statement. The decision variables and their initial and optimal values are given in Table S. Unlike problem 3. this ammonia process has a single loop design with 3 flash units. The unit operation and cost blocks for the reactor are taken from Chapters 9 and 10. respectively, of the FIOWTRAN manual. To make the problem more interesting, feed rates were chosen as decision variables and s constraint was imposed on the flow rate of the ammonia product. This type of constraint can be treated in a straightforward manner by all four of the algorithms compared.

Because of problems with error termination in FLOWTRAN. we suppressed the Broyden option for the first iteration in the EBOPT run. Even so, this run. as seen

from Table 1. required only 47% of Hie effort i»»«i ihe infaasihle path algorithm noeUed Broydon iterations were applied «n•> the J«1 4ih. bin A>n *nd 9t* SOP iteration However, the accelaianon* after t»« 3«d 4ih and 5ih iioiiioni were not effective (and also did not lead to closer pomisl because they ted to using more than 5 Broyden iterations without satisfying the ratio lest. This illustrates the difficulty of converging this flowsheet from intermediate points.

Similar, but more pronounced results were encountered with the CFV algorithm. Here the convergence algorithm was unable to converge the flowsheet for the first three SOP Iterations. For these points the maximum of 30 Wegstein iterations was exceeded end. consequently. CFV required a lot of computational effort. On the other hand, the IPH algorithm did very well for this problem. Because it uses a fixed number of recycle iterations at intermediate points, the progress of the optimization was better than IP. but none of the convergence problems encountered with CFV. or. to a lesser extent, with EBOPT. were observed here. Also for this problem there were no apparent difficulties with line search failures.

In summary, partial convergence of the flowsheet at Intermediate optimitation iterations led to better results on all of the recycle optimization problems then with either the IP or CFV algorithms. However, as shown in section 3. cere must be taken to implement this strategy properly. Therefore, this study illustrates the potential of the EBOPT strategy for flowsheet optimization, although further work may be required to tune the algorithm for specific problems.

REFERENCES

(i)Ballmen. S.K end J.L. Caddy. "Optimization of Methanol Process by Flowsheet Simulation". I&EC Pioc. Des. Dev.. 16, 3. p. 337. (1977)

(2]Berna. T.J.. MK Locke and AW. Westerberg, "A New Approach lo OptimaAlton of Chemical Processes". AICh€ J.. 20. 1. p. 37. (1980)

(3)Biagler. L.T.. "Improved Infeasible Path Optimization for Saquantial Modulars Simulators. Part I: The Interface." Comp. and Cham. Eng.. 9. 3. p. 245. (1985)

(4]Biaglar, L.T. and J.E. Cuihrall. "Improved Infeasible Path Optimization for Sequential Modulars Simulators. Part II: The Optimization Algorithm." Comp. end Cham. Eng. 9. 3. p. 257. (19851

(S]Biegler. L.T. and R.R. Hughes. "Infeasible Path Optimization of Sequential Modular Simulators". AIChE J.. 28. 6. p. 994. (1982)

(6]Biegler. L.T. and R.R. Hughes. "Optimization of Propylene Chlorination Process: A Case Study Comparison of Four Algorithms". Comp. and Cham Eng.. 7. 5. p. 645 (1983)

(7]Biegler. L.T. and R.R. Hughes. "Feasible Path Optimization for Sequential Modular Simulators". Comp. and Chem. Eng.. 9. 4. p. 379. (1985)

[8]Challand. T.B.. "Computerized Optimization of Complete Process Flowsheets". Chem. Eng. Prog.. 79. 6. p. 65. (1983)

[9]Chen. H-S and M.A. Stadtherr. "A Simultaneous Modular Approach to Process Flowsheeting and Optimization". AIChE J.. 31. 11. p. 1843. (1985)

[I0]0ennis. J.E. and J.J. More. "Quasi-Newton Methods. Motivation and Theory'. SIAM Review. 19. 1. p. 46 (1977)

(1 DFnedman. P. and K.L Pmder. "Optimization of a Simulation Model of Chemical Plants". I&EC Proc. Oes. Oev.. 11. 4. p. 512. (1972)

(i2]Gaines. LO. and J.L. Gaddy. "Process Optimization by Flowsheet Simulation". I&EC Proc. Oes. Dev.. 15. 1. p. 206 (1976)

[i3)Griffith. R.E. and R.A. Stewart. "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems". Mgt. Scl.. 7. p. 319 (1961)

[M]Han. S-P. "A Globally Convergent Method for Nonlinear Programming". J. OpL and Appl.. 22. 3. p. 297 (1977)

(15] Isaacson. R.A_ Ph.0 Thesis. University of Wisconsin. Madison (1975)

I I6]jirapongphan. S. Sc.O Thesis. Massachusetts Institute of Technology. Cambridge. MA (1980)

(I7)Kai|aluoio. S. "Process Optimization by Flowsheet Simulation". Technical Rosearch Center of Finland. Publication #20. (1984)

(I8)Kisala. T.P. Sc.O Thesis. Massachusetts Institute of Technology. Cambridge. MA (1985)

(!9)Meiceife. S.R. and J.O. Perkins. "Information Flow in Modular Flowsheeting Systems". Trans. I. Cham. E.. 66. p. 210. (1978)

(20)Parker. A.L. and R.R. Hughes. "Approximation Programming of Chemical Processes". Comp. and Chem. Eng.. 5. 3. p. 123. (1981)

(2i)Perkins. J.D.. "Efficient Solution of Design Problems Using a Sequential Modular Flowsheeting Programme". Comp. end Chem. Eng.. 3. p. 375 (1979)

(22]Powell. M.J.D.. "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations". 1977 Dundee Conf. on Num. Analysis. (1977)

(23)Rosen. E.M. and AX. Pauls. "Computer Aided Process Design: The FLOWTRAN System". Comp. and Cham. Eng.. 1. 1. p. 11 (1977)

(24]Seader. J.D.. W.D. Saidar and AX. Pauls. *FLOWTRAN Simulation - An Introduction.* 2nd edition. CACHE Corp. (1977)

IÅIIIJ I fttsi.li>. 11 V.il tonK llpl liaf z.il ion Strategies

No. Study Case	1III|-<1S|I I | c I'-Uli (IP)	rmirr	1I-ii	OV
1. Disti 1 IJUUII				
Objective function	7.33361	—	- -	
No. of iterations	7	—	—	
CPU Time (second)	111.09	—	—	
STE's (6.74 seconds/STE)	16.48	—	—	
2. CJVCCC problea				
Objective function	—	—	—	–
No. of iterations	4	6	—	13
CPU Time (second)	461.27	145.19	—	36.86
STE's (36.83 seconds/STE)	12.52	3.94	—	1.00
3. Ammonia A				
Objective funcClon	-3.67128	-3.67115	-3.966**	-3.67097
No. of Iterations	11	7	8	6
CPU Time (second)	1008.79	652.36	976.06	702.41
STE's (68.27 seconds/STE)	14.78	9.56	14.29	10.29
Ranges for x_1, x_2		0.036 *»j * 0.55 0.0018 *\ * 0.0186		
4. MCB				
Objective function	-0.898925*	-0.897342	-0.91142**	-0.89525†
No. of Iterations	12	7	12	10
CPU Tlae (second)	212.78	120.08	241.02	642.21
STE's (19.83 seconds/STE)	10.73	6.06	12.15	32.33
Ranges for x_1, x_2		x_1 > 0.268 x_2 > 0.078		
5. Amonla B				
Objective function	-24.9421	-24.9270	-24.9380	-24.9277
No. of Iterations	26	11	7	8
CPU Ti»e (second)	3604.56	1672.57	922.62	2480.77
STE's (215.90 seconds/STE)	16.70	7.75	4.27	11.49
Ranges for x_1, x_2		x_1 < 0.443 0.0074 s*2 & 0.0126		

* «»iraCeJ at [n[.asible pui"t due to line search InlUare

TAHU. 7. Cavett Problem

Values of Streams	Feed	Solution		Starting pt	
		/1	/2	/1	7?
N_2	358.2	391.2	J3.2	37.2	3i8.2
H_2S	339.4	508.9	263.6	200	339.4
CH_4	2995.5	3602.5	646.0	548.3	2995.5
C_2H_6	2395.5	3499.9	1524.2	1199.0	2395.5
C_4	604.1	752.2	636.6	523.0	604.1
[illegible]	1129.9	1212.3	1149.3	1082.4	1129.9
NC_6	1764.7	1812.6	1771.0	1736.6	1764.7
« 7	2606.7	2636.3	2607.6	2589.6	2606.7
NC_8	1844.5	1852.9	1844.0	1839.8	1844.5
NC_{10}	831.7	832.4	831.6	831.3	831.7
NC_{11}	1214.5	1215.0	1214.4	1214.2	1214.5
T	120	118.3	71.55	120	120
P	49	49	13	49	49

TA4HJ.). H.M.II ul AMMOIIIa A Problea

	Item	Optimum	Starting Pt.
A.	Objective Function	-3.67128	-3.23W
It	Design Variables		
	1. Inlet Temp of Kcactor	147.226	180
	2. press of Kcactor	29SO	3000
	3. Inlet Tea? of High Pressure Flssh	-24.0	-24
	4. Purge Frsctlon	0.0802432	0.10
	3. Conversion of Nitrogen (X)	45.0	41.0
C.	Tear Variables		
	I. Flowrate		
	H_2	2193.02	2170.
	N_2	568.426	612.0
	NH_3	30.1334	16.3
	Ar	85.7868	70.5
	CH_4	104.604	125.0
	2. Temperature	289.883	215.0
	3. Pressure	2950.0	3000.0

TAMO. 4. M.»..|| o(Ht >\ I'roblea

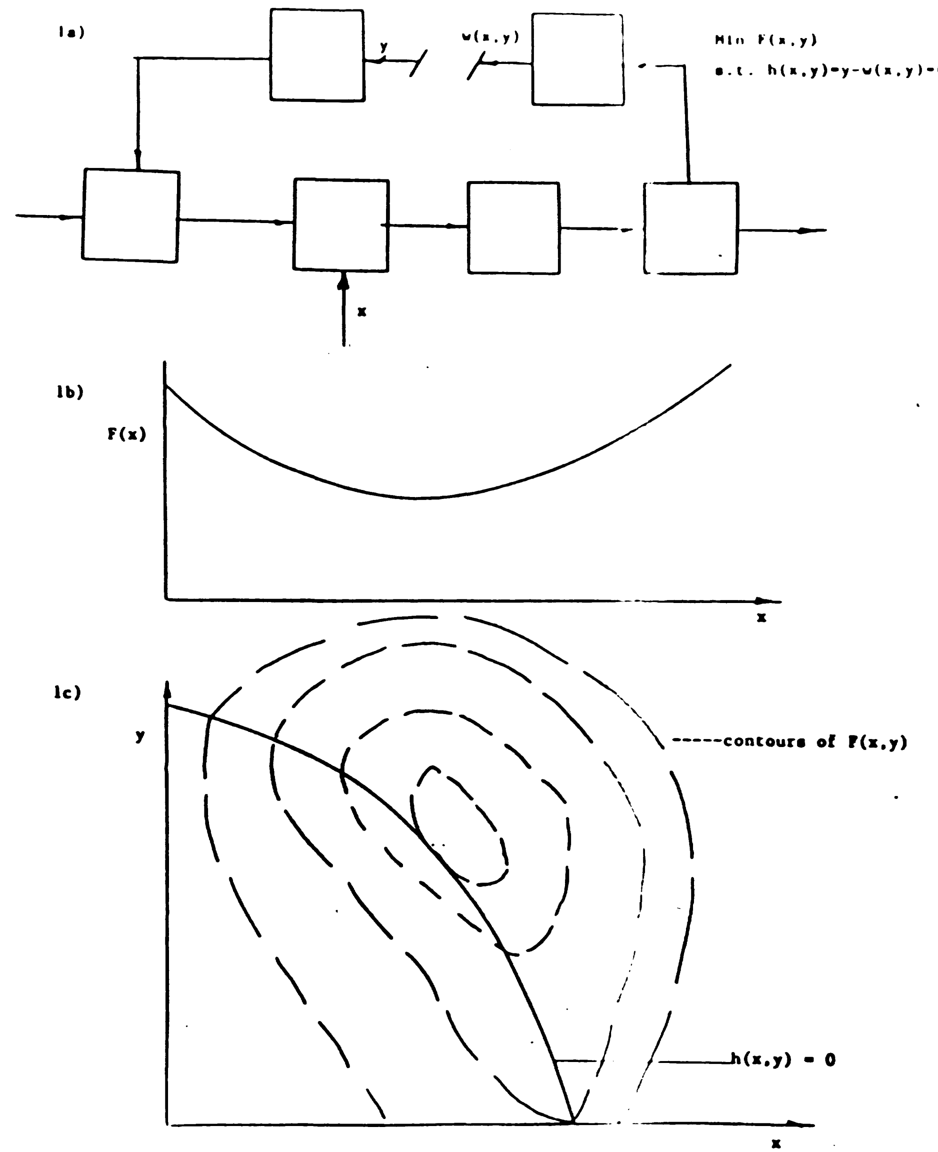
No.	Item	Oft Itow	Starting Pt tut lath	ILL.!:1II I
A.	Objective Function	-0.898925	0.W06/	-0 MO)1)
B.	Design Variables			
	1. lottos press, of Absorber (psls)	32.0401	32.0	32.0
	2. Top press, of Absorber (psls)	31.0401	31.0	31.0
	3. Spile Fraction of tecycle	0.377398	0.33	0.33
	4. Split Frsctlon of Withdrawal	0.622602	0.67	0.67
	5. In1st T««p. of Flash	227.422	270.0	270.0
	6. Outlet Tee?, of Hest Exchanger	100.0	120.0	120
C.	Tear Variables			
	Flowratcs:			
	1. HC1	0.0	0.0	0.0
	2. Benzene	0.623078	0.0	0.0
	3. MCI	82.1304	80.0	90.0
	Temperature	100.00	120.0	100.00
	Pressure	50.0	50.0	50.0

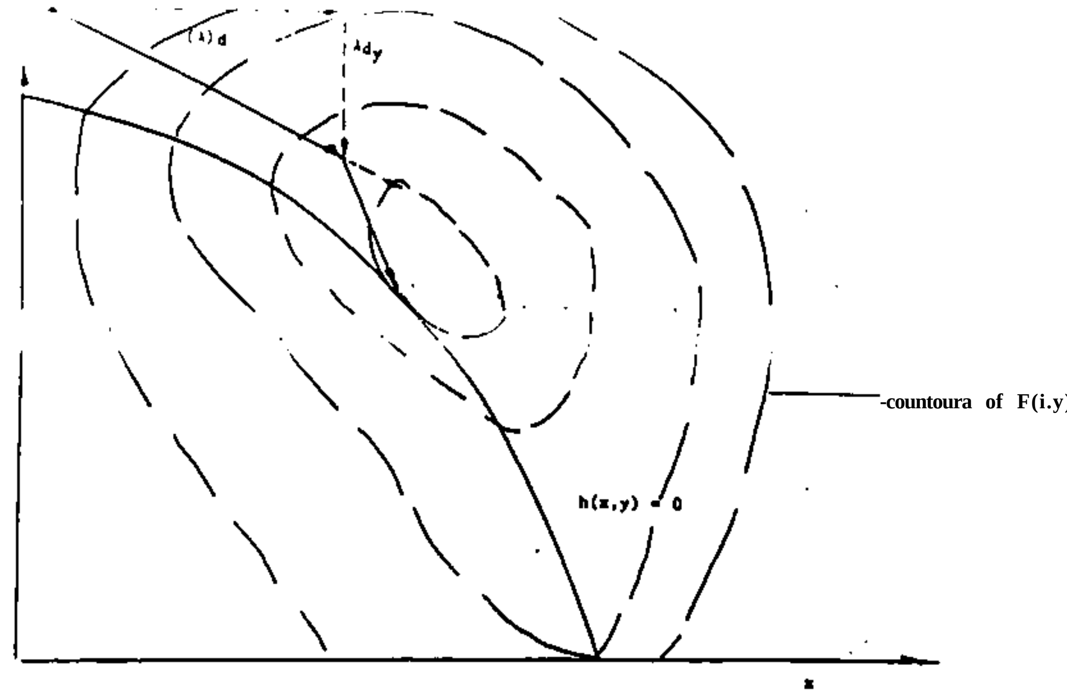
TABLE 5. Result of Ammonia B Problem

No.	Item	Optimum	Starting Pt.
A.	Objective Function	-24.9421	-20.659
B.	Design Variables		
	1. Inlet Temp. of Reactor	400.0	400.0
	2. Inlet Temp. of 1st Flash	65.0	65.0
	3. Inlet Temp. of 2nd Flash	35.0	35.0
	4. Inlet Temp. of Recycle Compressor	75.3139	107.0
	5. Purge Fraction (%)	0.814030	1.0
	6. Inlet Press. of Reactor	1984.69	2000
	7. Flowrate of Feed 1	2626.66	2632
	8. Flowrate of Feed 2	688.206	1648
C.	Tear Variables		
	1. Flowrate		
	N_2	1303.88	1648.0
	H_2	3921.21	3676.0
	NH_3	551.760	424.9
	Ar	183.708	143.7
	CH_4	2095.34	1657
	2. Temperature	75.3139	60
	3. Pressure	1901.49	1930

Figure 1

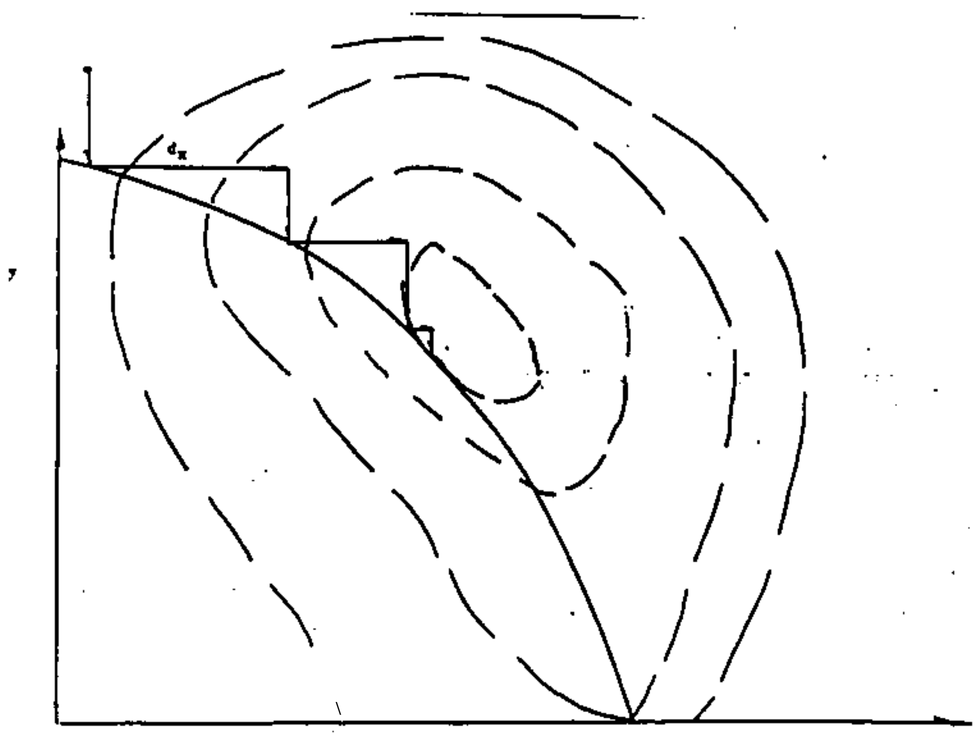
Simple Flowsheet Optimization ...em





Figurt 3a .

Inftaalblt PathOP) Optimisation Strategy



Flgurt 2

Black Box Optimisation **Strategy**

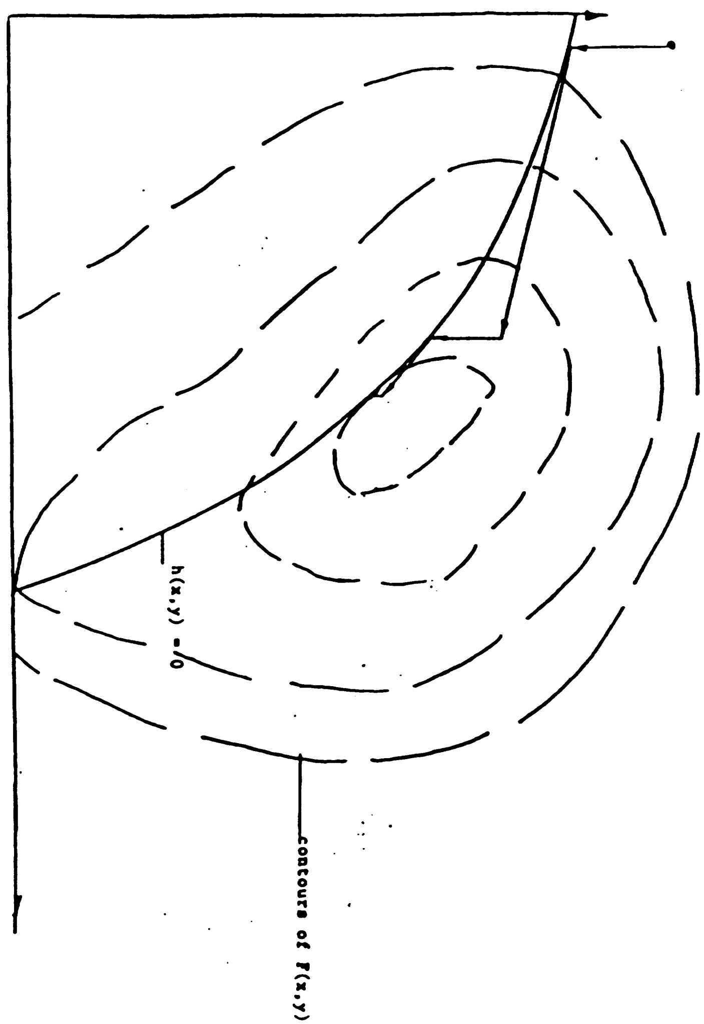


Figure 3b

Feasible Variant Optimization Strategy

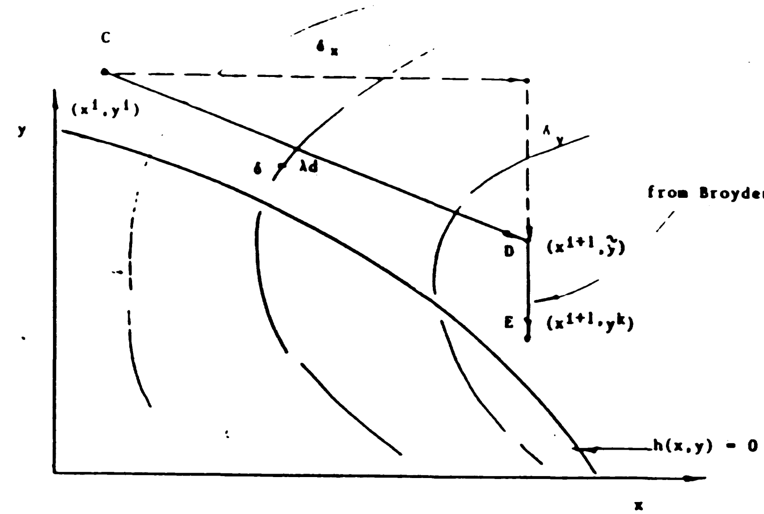


Figure 4

Partially Converged Strategy
for EBOPT

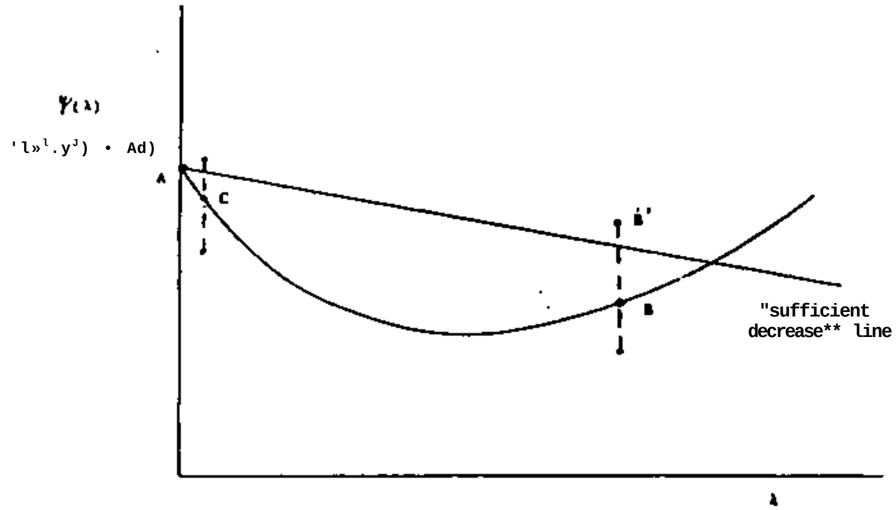


Figure 5

Source of Line Search Failures
with Partial Convergence

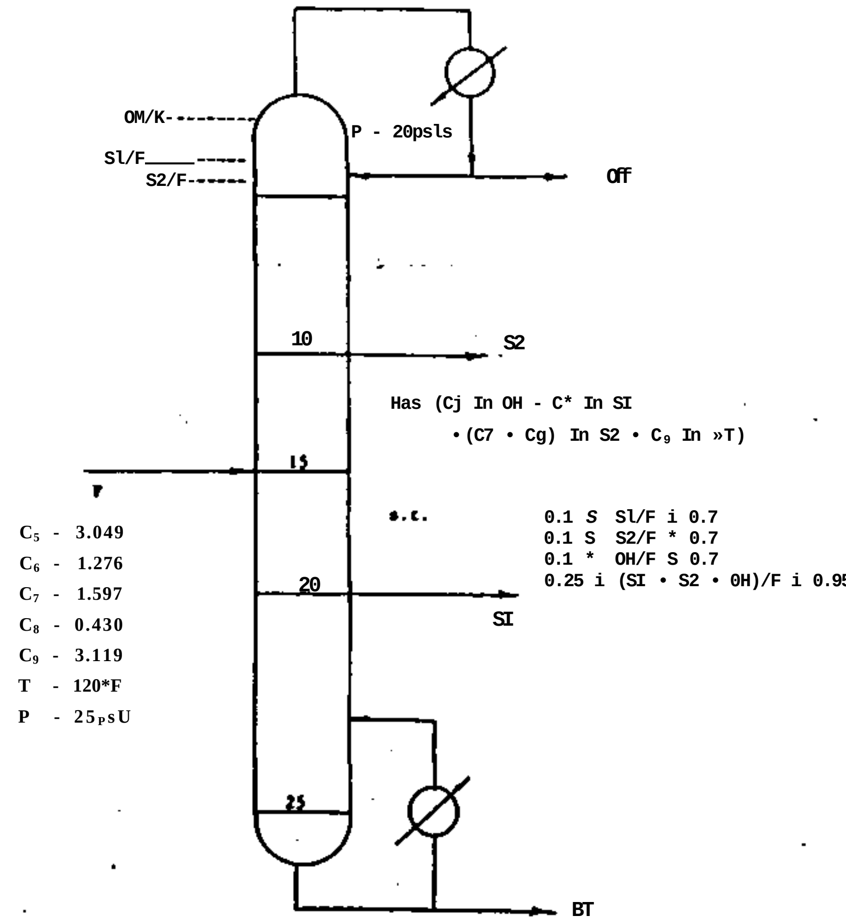


Figure 6

Dlstlllstlon CelusB Opclmlsacloo

Initial

ObJ. Fen - 6.M2
Sl/F - 0.2
S2/F - 0.2
OH/F - 0.3

Optimum

ObJ. Fen - 7.554
Sl/f - 0.1
S2/F - 0.2499
OM/F - 0.3353

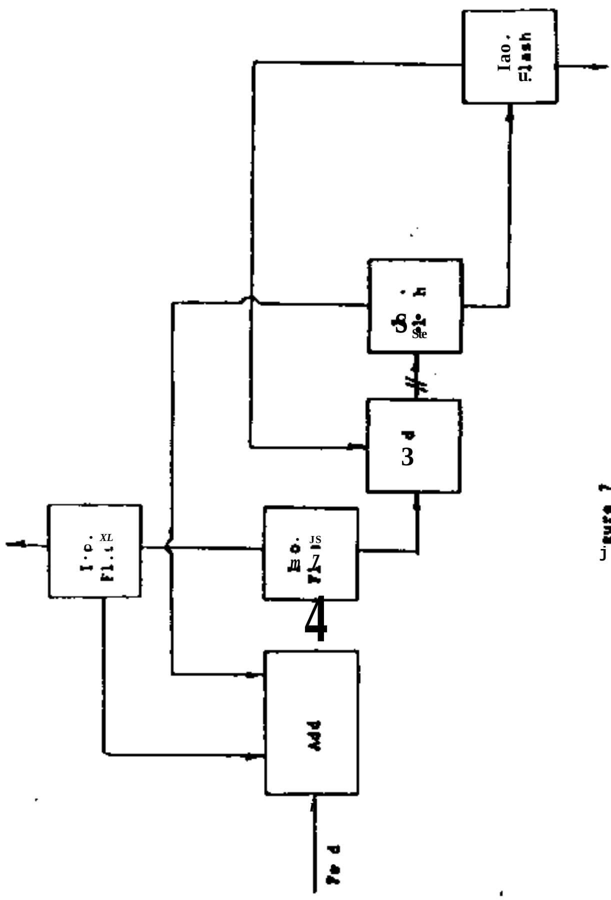
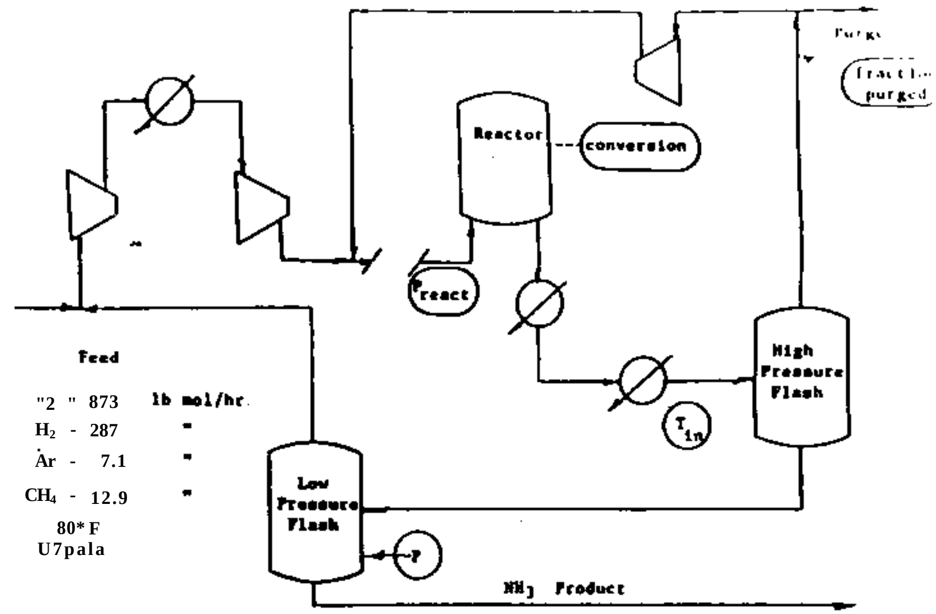


Figure 7
Partial Cavett Problem - 11 components
(see ... & Paules)
- Converge with Newton's method (IP)
- Converge with Broyden's method (ZBOPT)
- Degrees of freedom

Figure 8
Ammonia Process A - due to Parker & Hughes (1981)



Min [Venture Profit over 10 years t 15Z)

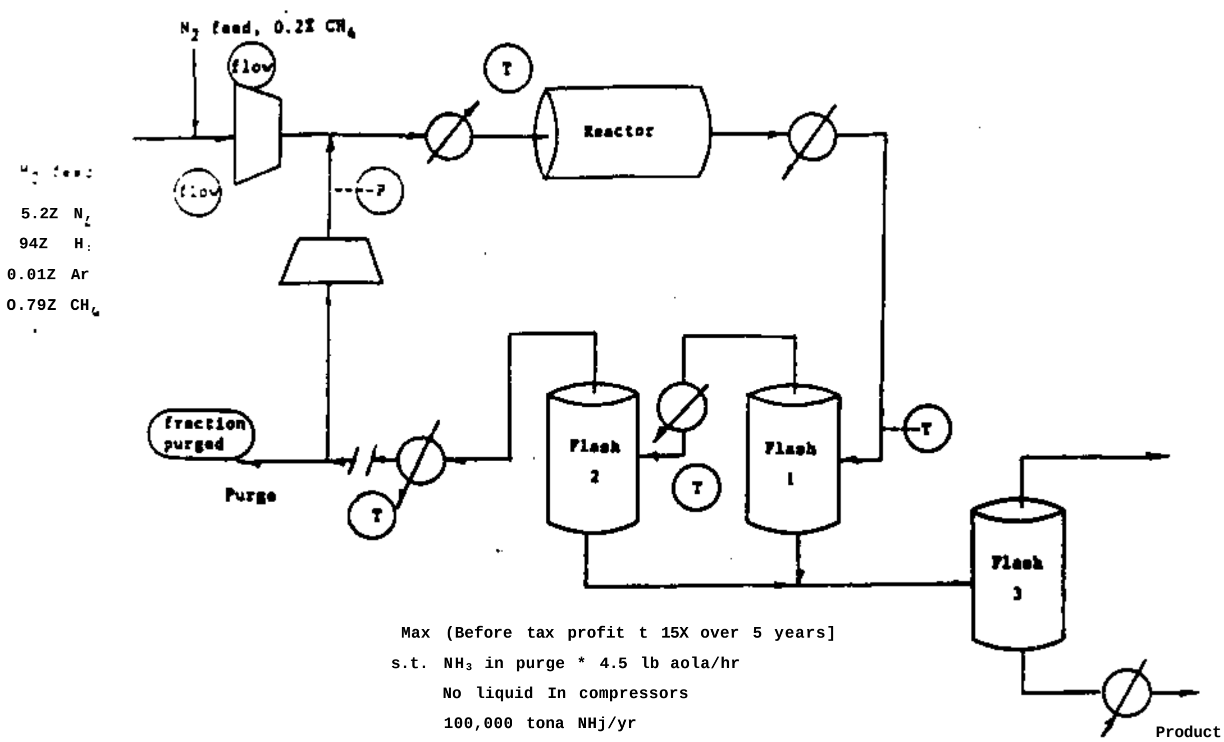
s.t. HH3 purity la product 2 95Z

MH3 In purge S 1.5 ltmol/hr

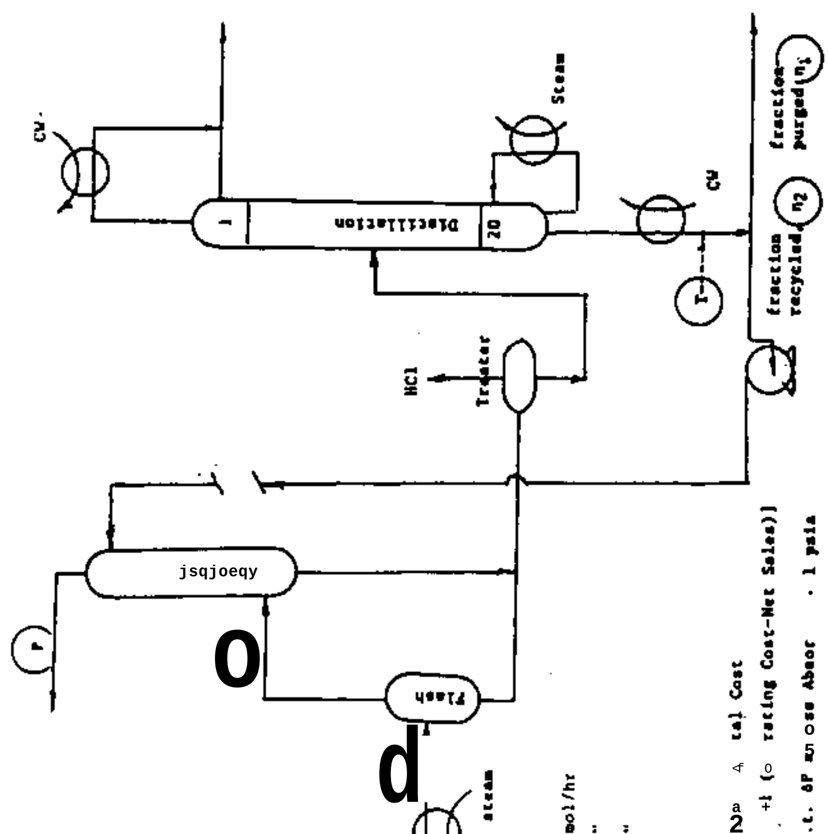
teactor teap i 1000*P

Vapor In coapreaaora I 99X

Figure 10
Ammonia Process



Max (Before tax profit t 15% over 5 years]
s.t. NH_3 in purge * 4.5 lb mole/hr
No liquid In compressors
100,000 tons NH_3/yr
NH_3 Product Purity ≥ 99.9%
1.8 ≤ H_2/N_2 In ≤ 3.3 combined feed



Feed
HCl 10 lb mol/hr
Benzene 40
MCB 50

Figure 9
Monochlorobenzene Process Optimization