Universidade Federal do Rio de Janeiro
Escola de Química

Programa de Pós-Graduação em
Tecnologia de Processos Químicos e Bioquímicos

Automatização do Diagrama de Fontes de Água

Flavia Pellegrini Naice

Rio de Janeiro

2015

Flavia Pellegrini Naice

Automatização do Diagrama de Fontes de Água

Dissertação de Mestrado apresentada ao Programa de Tecnologia de Processos Químicos e Bioquímicos da Escola de Química da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências.

Orientadores:

Fernando Luiz Pellegrini Pessoa, D.Sc.
Eduardo Mach Queiroz, D.Sc.

Rio de Janeiro
Maio de 2013

N155a

Naice, Flavia Pellegrini
Automatização do Diagrama de Fontes de Água / Flavia Pellegrini Naice. -- Rio de Janeiro, 2015.
90 f.

Orientador: Fernando Luiz Pellegrini Pessoa.
Coorientador: Eduardo Mach Queiroz.
Dissertação (mestrado) - Universidade Federal do Rio de Janeiro, Escola de Química, Programa de Pós-Graduação em Tecnologia de Processos Químicos e Bioquímicos, 2015.

1. Diagrama de Fontes de Água. 2. Minimização de Efluentes Aquosos. 3. Reuso de água. I. Pessoa, Fernando Luiz Pellegrini , orient. II. Queiroz, Eduardo Mach, coorient. III. Título.

Flavia Pellegrini Naice

AUTOMATIZAÇÃO DO DIAGRAMA DE FONTES DE ÁGUA

Dissertação de Mestrado apresentada ao Programa de Tecnologia de Processos Químicos e Bioquímicos da Escola de Química da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências.

Orientada por:

_____

Fernando Luiz Pellegrini Pessoa, D.Sc.

_____

Eduardo Mach Queiroz, D.Sc.

Aprovada por:

_____

Lídia Yokoyama, D.Sc.

_____

Reinaldo Coelho Mirre, D.Sc.

_____

Rogério Mesquita de Carvalho, D.Sc.

Rio de Janeiro

Maio de 2015

Agradecimentos

Aos meus queridos pais, Paulo e Madalena, pelo amor e o incentivo.

Ao meu marido, Leonardo, pela paciência e a compreensão.

Aos meus orientadores, os Profs. Fernando Luiz Pellegrini Pessoa e Eduardo Mach Queiroz pela confiança e pela atenção durante a realização desse trabalho.

Aos amigos do grupo GIPQ.

E todos aqueles que contribuíram de alguma forma para a realização desse trabalho.

Resumo da Dissertação de Mestrado apresentada ao Curso de Pós-Graduação em Tecnologia de Processos Químicos e Bioquímicos da Escola de Química/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências, com ênfase na área de Engenharia de Processos.

AUTOMATIZAÇÃO DO DIAGRAMA DE FONTES DE ÁGUA

Flavia Pellegrini Naice
Maio, 2015

Orientadores: Prof. Fernando Luiz Pellegrini Pessoa, D.Sc.
Prof. Eduardo Mach Queiroz, D.Sc.

A Integração Mássica surgiu em resposta à necessidade do setor industrial em reduzir o consumo de água no processo produtivo. Uma das maneiras de reduzir esse consumo é através do reuso dos efluentes das operações que usam água. A minimização dos efluentes aquosos é um problema de otimização que tem por um dos objetivos a síntese da rede de transferência de massa que corresponde ao menor consumo de água primária. O Diagrama de Fontes de Água (DFA) é um procedimento algorítmico-heurístico que possibilita encontrar a especificação da meta de consumo mínimo e, simultaneamente, a rede de transferência de massa. O procedimento pode ser aplicado a problemas com um ou com múltiplos contaminantes. Apesar de se tratar de uma metodologia de fácil aplicação dos cálculos manuais, ela se torna muito laboriosa quando os problemas atingem grandes dimensões. Com o intuito de aprimorar o uso do DFA, foi desenvolvido um *software*, denominado MINEA 2.0, baseado no algoritmo para o caso de máximo reuso. Para tal, foi escolhida a linguagem C# da plataforma .NET, a qual possibilita o desenvolvimento orientado a objeto, permitindo uma construção componentizada que favorece futuras evoluções do *software*. Exemplos que são usualmente encontrados na literatura foram executados no *software* e os resultados encontrados foram condizentes com os resultados demonstrados manualmente, indicando que o MINEA 2.0 apresenta boa *corretude* funcional.

Palavras-chave: Diagrama de Fontes de Água, minimização de efluentes aquosos, reuso de água.

Abstract of Thesis presented to Curso de Pós-Graduação em Tecnologia de Processos Químicos e Bioquímicos (EQ/UFRJ) as a partial fulfillment of the requirements for the degree of Master of Science, with emphasis on Chemical Engineering Process Design.

Automatization of Water Source Diagram

Flavia Pellegrini Naice
Maio, 2015

Advisors: Prof. Fernando Luiz Pellegrini Pessoa, D.Sc.
Prof. Eduardo Mach Queiroz, D.Sc.

Mass Integration was a response to the industrial sector need of a reduction of production process water consumption. One way of doing it is through reuse of streams from water-using operations. The minimization of water flow rate is an optimization problem that has as one of its goals the synthesis of the mass exchange network corresponding to a least primary water consumption. The Water Source Diagram (WSD) is an algorithmic-heuristic procedure that allows finding the specification for the minimum consumption target and, simultaneously, the mass exchange network. The procedure is applicable to single or multiple contaminants problems. Despite being an easy usage methodology when calculating manually, it can be laborious when these problems evolve into larger dimensions. With the goal to improve WSD usage, a *software* was developed – MINEA 2.0 – based on the maximum reuse algorithm. For such, C# language from .NET platform was chosen, as it allows object-oriented development, permitting a component-oriented construction, enabling future *software* evolutions. Examples commonly found in literature were executed on the *software* and the presented results were consistent to the ones found manually, pointing that MINEA 2.0 presents good correctitude.

Keywords: Water Source Diagram, wastewater minimization, water reuse

SUMÁRIO

Capitulo 1 – Introdução 1
1.1. Objetivo do trabalho e apresentação dos capítulos 8
Capítulo 2 – Revisão Bibliográfica..... 10
2.1. *Software*s Desenvolvidos 16
Capítulo 3 – Diagrama de Fontes de Água 24
3.1. Presença de um contaminante..... 26
3.2. Presença de múltiplos contaminantes..... 32
Capítulo 4 - O *software* MINEA 2.0 40
Capítulo 5 - Exemplos de Aplicação..... 48
5.1. Wang e Smith (1994) – Máximo reuso 48
5.2. Wang e Smith (1994) - Múltiplas Fontes de Água 49
5.3. Wang e Smith (1994) - Regeneração com reciclo..... 50
5.4. Wang e Smith, 1994. Máximo Reuso 52
5.5. Doyle e Smith, 1997 – Máximo Reuso. 55
5.6. Relvas *et al* (2005) – Máximo Reuso 58
5.7. Relvas *et al* (2005) – Regeneração com Reciclo 58
5.8. Prakotpol e Srinophakun (2004) – Máximo reuso 59
Capítulo 6 - Conclusão..... 62
Referências Bibliográficas..... 63
Anexo A.1..... 68
Anexo A.2..... 70

Lista de Figuras

Figura 1.1 - Minimização do consumo de água primária via (a) reuso; (b) regeneração com reuso; (c) regeneração com reciclo 6
Figura 3.1 - Equipamento de Transferência de Massa (ETM). CP_i é a corrente de processo que entra no ETM; CP_f é a corrente de processo que sai do ETM; Ca_i é a corrente do agente extrator que entra no ETM; CA_f é a corrente do agente extrator que sai do ETM 25
Figura 3.2 – Representação do Passo 1 do Diagrama de Fontes de Água 27
Figura 3.3 – Representação das operações no Diagrama de Fontes de Água (Passo 2) 28
Figura 3.4 - Representação das operações no Diagrama de Fontes de Água (Passo 3) 29
Figura 3.5 – Representação do Diagrama de Fontes de Água. 30
Figura 3.6 – Rede de Transferência de Massa para a opção de máximo reuso correspondente ao diagrama da Figura 3.5..... 31
Figura 3.7 – *Workflow* do Diagrama de Fontes de Água de sistemas com um contaminante (FEA é a Fonte Externa de Água; FIA é a Fonte Interna de Água; TM é a transferência de massa para uma dada operação em um dado intervalo de concentração)..... 31
Figura 3.8 – Diagrama de Fontes de Água para o problema exemplo 37
Figura 3.9 – Rede de Transferência de Massa corresponde ao diagrama da.. 38
Figura 3.10 – *Workflow* do Diagrama de Fontes de Água para sistemas multicontaminantes (Gomes *et al*., 2013 – Adaptado)..... 39
Figura 4.1 – Primeira tela de inserção de dados do MINEA 2.0..... 43
Figura 4.2 – Segunda tela de inserção de dados do MINEA 2.0..... 44
Figura 4.3 – Caixa de Diálogo do botão “Ajuda” pertencente à Caixa de Seleção da operação de referência..... 45
Figura 4.4 - Caixa de Diálogo do botão “Ajuda” pertencente à Caixa de Seleção do contaminante de referência 45
Figura 4.5 – Terceira tela do MINEA 2.0 46
Figura 4.6 – Imagem do Diagrama de Fontes de Água criada pelo MINAE 2.046
Figura 4.7 – Tela do “Relatório” com as informações sobre o diagrama e sobre as operações..... 47

Figura 5.1 – Diagrama de Fontes de Água referente ao problema 5.2.1 49
Figura 5.2 – Rede de Transferência de Massa correspondente ao diagrama da 49
Figura 5.3 – Diagrama de Fonte de Água do exemplo 5.2. 50
Figura 5.4 – Primeira tela de inserção de dados do MINEA 2.0 considerando regeneração com reciclo 51
Figura 5.5 – Diagrama de Fontes de Água para o problema 5.3..... 51
Figura 5.6 – Diagrama de Fontes de Água do exemplo 5.4 53
Figura 5.7 – “Relatório” correspondente ao diagrama da Figura 5.6..... 54
Figura 5.8 – Diagrama de Fontes de Água para o exemplo 4.5 56
Figura 5.9 – “Relatório” correspondente ao diagrama da Figura 5.8..... 57
Figura 5.10 - Diagrama de Fontes de Água referente ao exemplo 5.6..... 58
Figura 5.11 - Diagrama de Fontes de Água do exemplo 5.7..... 59
Figura 5.12 – Diagrama de Fontes de Água referente ao exemplo 5.8..... 60

Lista de Tabelas

Tabela 3.1 – Dados do problema exemplo (Wang e Smith, 1994) 26
Tabela 3.2 – Dados de limites para o Problema Exemplo (Wang e Smith, 1994) 32
Tabela 3.3 - Valores ajustados das Concentrações de B na Operação 1 34
Tabela 3.4 - Matriz das Concentrações Ajustadas 36
Tabela 3.5 - Problema Exemplo baseado no Contaminante de Referência 37
Tabela 5.1 - Dados do problema exemplo 5.1 48
Tabela 5.2 – Dados do problema exemplo 5.4 52
Tabela 5.3: Dados do problema exemplo 4.2.5 55
Tabela 5.4 **– Dados do problema 5.6** 58
Tabela 5.5 **– Dados do problema 4.2.8** 59

Capitulo 1 – Introdução

É incontestável a importância da água para a humanidade. Durante milênios o homem utilizou os recursos hídricos para diferentes propósitos. Além de ser um elemento indispensável para a manutenção de todos os organismos vivos e dos ecossistemas, a água é essencial para diversas atividades como a agricultura, a pecuária e os processos industriais.

A água foi um insumo decisivo na História do mundo: sua qualidade e quantidade, a distribuição e a sazonalidade ditavam os pontos de colonização. Não foi ao acaso que as grandes civilizações da Antiguidade nasceram às margens de algum grande rio. Nesse período, preludiava o gerenciamento dos recursos hídricos e, assim, os métodos de irrigação e drenagem foram aperfeiçoados, iniciou-se a implantação dos sistemas de elevação de água e a construção de canais e poços. O sucesso do gerenciamento dos recursos hídricos propiciou a produção de alimentos e, aliado às terras férteis, modelava-se uma nova estrutura social e o aumento da população. De forma geral, o gerenciamento da água possibilitou transformações históricas no desenvolvimento do modo de vida das sociedades.

Com a Revolução Industrial nos anos de 1800, o crescimento populacional acompanhava o rápido avanço industrial, as cidades começavam a ser urbanizadas e, consequentemente, a demanda por água aumentou sensivelmente a fim de atender aos novos serviços, às indústrias e à crescente população. No período pós Segunda Guerra (1951-1960), o mundo apresentava avanços tecnológicos e científicos que possibilitaram o crescimento dos setores produtivos; a demanda por água já era quatro vezes maior que as décadas anteriores devido à expansão das terras irrigadas, ao rápido desenvolvimento industrial e à intensa construção de reservatórios (SHIKLOMANOV, 1998).

Durante muitos anos, o impacto das atividades do homem sobre os recursos hídricos foi insignificante e de caráter local. O processo de autodepuração dos corpos de água e a regeneração da água durante o seu ciclo eram eficientes possibilitando a manutenção da qualidade da água doce. Entretanto, nas últimas décadas, a eficiência de recuperação dos corpos hídricos foi exaurida.

Dados da Organização das Nações Unidas mostram que, apesar da água ser um recurso abundante no planeta, estima-se que o volume de água doce ocupe somente 2,5% do volume total disponível para uso. Dessa fração, aproximadamente 30% corresponde a fatia de água doce disponível para o consumo humano, valores que descredenciam a ideia de que a água é um recurso inesgotável (ONU Water).

Além da disponibilidade variar com o tempo e com o espaço, a utilização da água leva à contaminação pelos mais diversos tipos de substâncias desenvolvidas e utilizadas pelo homem, o que dá origem aos processos de poluição dos recursos hídricos e contribui para a redução da disponibilidade da água. Os problemas mais sérios de poluição referem-se ao efluentes industriais que apresentam em sua composição os mais variados tipos de contaminantes (MIERZWA, 2002).

No século XX, as questões relacionadas ao meio ambiente entraram na pauta de debate de todas as esferas da sociedade. Os impactos ambientais resultantes da vida cotidiana tornaram-se tema da agenda política de diversos governos e projeções alarmantes referentes à poluição ambiental, à escassez dos recursos naturais e ao aquecimento global motivaram muitos estudos acerca de possíveis soluções e medidas de preservação ambiental. Desde a década de 1970, discussões sobre essas questões vem ganhando cada vez mais espaço.

No ano de 2014, o sudeste brasileiro enfrenta a pior crise hídrica dos últimos 84 anos. Em janeiro de 2015, o Sistema Cantareira, que é responsável pelo abastecimento de cerca de 9 milhões de pessoas que vivem na capital e na região metropolitana do Estado de São Paulo, alcançou 5,1% do nível operacional (SABESP, 2015); no mesmo período, o reservatório Paraíbuna, o principal sistema que abastece o Rio de Janeiro, atingiu a reserva técnica (NOS,

2015). O governo de São Paulo decidiu restringir a irrigação nas bacias do Alto Tietê e do Piracicaba, Capivari e Jundiaí, medida que afeta diretamente o setor agrícola e a economia, gerando limitações no abastecimento e elevação dos preços dos alimentos. A crise hídrica vivenciada pelo sudeste brasileiro fomentou novamente discussões a respeito do desperdício de água, das altas taxas de cobrança pelas concessionárias e do impacto ao meio ambiente.

Além das questões ambientais, a falta de água ainda pode ocasionar um efeito negativo na economia do país: a baixa nos reservatórios pode afetar o setor hidroelétrico; alta dos índices de inflação; racionamento hídrico e energético; elevação nos preços dos alimentos e serviços.

Não são recentes as discussões ambientais relacionadas ao não desperdício de água, o fato de não ser um recurso infinito, o controle ou prevenção da poluição, o gerenciamento urbano dos cursos d'água, e os custos relacionados ao uso da água e ao tratamento de efluentes.

Diante desse cenário, tornou-se imperiosa a adoção de práticas sustentáveis e soluções ambientais que mitigassem os efeitos colaterais causados pelos setores produtivos, mais especificamente, o setor industrial que faz parte do escopo desse trabalho. As indústrias química e petroquímica utilizam grandes volumes de água em seus processos. A demanda de água de uma indústria está diretamente relacionada ao segmento industrial a que ela pertence e a sua capacidade de produção. O setor de alimentos, por exemplo, chega a utilizar 150.000L de água por tonelada de produção. A indústria de papel e celulose pode chegar a utilizar 1.000.000L de água por tonelada de produção e a indústria de petróleo e combustíveis utiliza na faixa de 500.000L/t de produção (MIERZWA, 2002 *apud* VAN der LEEDEN, TROISE e TODD, 1990). Como resultado do consumo de água, o setor industrial gera efluentes que contêm diversos tipos de contaminantes que, sem o devido tratamento antes do descarte, provocam danos ao meio ambiente.

Por muitos anos, a principal ação adotada pelas indústrias para resolver o problema gerado pelo descarte dos efluentes era sempre focada na prática de

tratamento de fim de tubo (*end-of-pipe wastewater treatment*) que consiste no tratamento dos efluentes ao final processo produtivo. Entretanto, quando os contaminantes são removidos durante o processo produtivo, o efluente pode ser reusado e/ou reciclado no processo. Isso produz um impacto direto no montante final de água utilizada tanto quanto no montante do efluente que atinge o final do processo (SAVELSKI e BAGAJEWICZ, 2000a).

A prevenção da poluição é um termo utilizado para designar as práticas que reduzem ou eliminam a geração de efluentes na fonte onde são gerados, através de modificações no processo de produção, promovendo o uso de substâncias com pouca ou nenhuma toxicidade, com o uso de tecnologias de conservação, reusando os materiais nas correntes de produção em vez de concentrá-los no efluente do processo (Agência de Proteção Ambiental dos Estados Unidos – EPA).

O controle na fonte inclui qualquer ação realizada no processo produtivo para reduzir a quantidade ou a toxicidade do efluente na fonte onde ele é gerado; o reuso/reciclo é a prática que reintroduz no processo produtivo as correntes concentradas de contaminante (EL-HALWAGI, 1998). Assim, as abordagens para o controle da poluição são voltadas para os problemas gerados no processo produtivo e, portanto, focadas no tratamento de efluentes distribuído no processo (*in-plant-treatment*), ultrapassando as habituais práticas de tratamento centralizado ou fim de tubo. De fato, reduzir ou eliminar a geração de espécies indesejáveis na fonte reduz significativamente os custos associados às práticas centralizadas (DUNN e EL-HALWAGI, 2003).

A minimização do consumo de água no processo industrial reduz os custos operacionais e a emissão de efluentes para o meio ambiente. Uma opção para reduzir o consumo de água e a geração de efluentes é realizar mudanças estruturais no processo ou nos equipamentos das operações. Alguns exemplos para essa opção são (GOMES, 2002):

- Aumento do número de estágios em processos de extração que utilizam água;

- Utilização de *spray-balls* para lavagem interna de vasos mais efetiva;
- Aumento do retorno de condensado nos sistemas de aquecimento (vapor) para reduzir as perdas de água e diminuir a demanda por tratamento de água de alimentação da caldeira;
- Aumento dos ciclos de recirculação do sistema de resfriamento;
- Melhoria da eficiência energética do processo para redução da demanda de vapor, causando a redução da geração de efluentes líquidos no sistema de vapor em função da consequente diminuição da necessidade de purga na caldeira, bem como das perdas de água no sistema de tratamento de água de alimentação da caldeira e das perdas em função da condensação.

Entretanto, existem algumas alternativas que não exigem grandes modificações no processo e que reduzem o consumo de água e a geração de efluentes. Essas alternativas são (RELVAS *et al*., 2008):

- Reuso: o efluente de algumas operações pode ser usado diretamente em outras operações. Talvez seja necessário que esse efluente seja combinado e/ou misturado com água fresca (primária) (Figura 1.1.a).

- Regeneração com reuso: o efluente de algumas operações é regenerado para remover os contaminantes de forma que ele possa ser reusado posteriormente em outra operação (Figura 1.1.b).

- Regeneração com reciclo: o efluente regenerado pode eventualmente ser redirecionado a operação de origem. Nos processos onde a complexidade é maior do que a apresentada na Figura 1.1.c, estratégias que combinam o reciclo e o reuso podem ser adotadas.

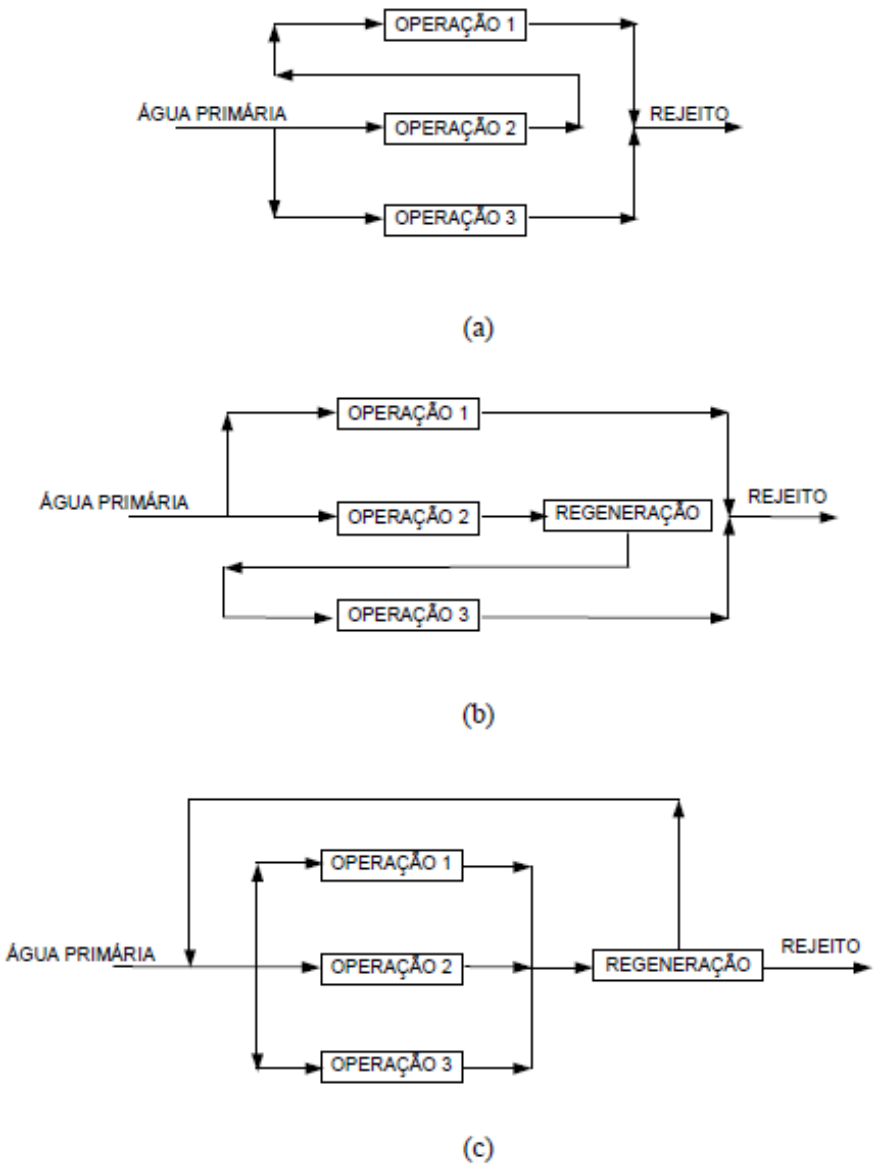

Figura 1.1 - Minimização do consumo de água primária via (a) reuso; (b) regeneração com reuso; (c) regeneração com reciclo

Nos últimos anos as indústrias vêm gradualmente migrando para a prática de controle de poluição na fonte, ou seja, no coração do processo.

Controlar na fonte implica em uma série de novas estratégias de controle de poluição que o projetista deve adotar; entretanto, essas novas práticas defrontam com alguns desafios encontrados na substituição da prática *end-of-pipe* pela *in-plant* (EL-HALWAGI, 1998):

- A fim de realizar quaisquer modificações no núcleo do processo, é inevitável que o projetista possua total compreensão da natureza integrada do processo, suas correntes e unidades operacionais;
- Mudanças nas unidades ou correntes operacionais podem influenciar na rentabilidade e no modo de operação do processo;
- Os objetivos ambientais não podem ser os carros-chefes para as mudanças na planta, pois o processo possui inúmeras restrições tecnológicas, econômicas e de segurança que também devem ser consideradas. A maior dificuldade é a conciliação entre todas as restrições e o fator ambiental;
- Além das estratégias de controle da poluição que envolvem a instalação de novos equipamentos, grande parte da solução envolve a utilização de equipamentos já existentes na planta industrial. Entretanto, não é trivial identificar essas oportunidades;
- As soluções encontradas para determinado processo normalmente não são replicáveis para outro processo. Diferenças sutis entre processos levam a soluções significativamente diferentes.

A partir da análise desses desafios, conclui-se que se torna necessária a aplicação de uma abordagem sistemática que transcende as particularidades do processo e encara o problema ambiental a partir de um ponto de vista holístico. Inserida nesse contexto, encontra-se a Integração de Processos, um campo pertencente a área dos processos químicos que proporciona um panorama único para enfrentar os desafios acima mencionados.

A minimização dos efluentes aquosos está diretamente relacionada a Integração de Processos, uma vez que esta última possui ferramentas que possibilitam a análise de problemas que envolvem a redução da geração de efluentes. Dentre essas ferramentas encontra-se o Diagrama de Fontes de Água, um procedimento algorítmico que possibilita determinar, usando regras heurísticas, o mínimo consumo de água de um processo industrial.

1.1. Objetivo do trabalho e apresentação dos capítulos

O objetivo principal desta dissertação de mestrado é desenvolver um *software* utilizando um *framework* e uma linguagem de programação que proporcionem um *software* mais robusto e estável para a aplicação do procedimento algorítmico Diagrama de Fontes de Água (DFA).

Como objetivos específicos têm-se:

1. Desenvolver uma interface amigável e intuitiva;
2. Centralizar as informações de entrada e de saída em um único local;
3. Desenvolver com baixo acoplamento, ou seja, com pouca dependência entre as instâncias, permitindo que evoluções futuras do *software* sejam feitas com baixo nível de retrabalho;
4. Garantir que o *software* resolva corretamente problemas de qualquer dimensão.

Existem diversas opções de aplicação do DFA. O caso de opção de máximo reuso (apresentado nesta dissertação) é a base do algoritmo e, por isso, a construção do *software* se manteve fiel ao procedimento DFA, tanto para o caso de sistemas com a presença de um contaminante quanto para o sistema multicontaminantes.

Esta dissertação está organizada em cinco capítulos, contando com o presente Capítulo de Introdução.

No Capítulo 2 são apresentados alguns conceitos fundamentais relacionados ao controle da poluição e à minimização dos efluentes aquosos. São apresentados alguns *software*s desenvolvidos na área acadêmica e são citados alguns *software*s e solvers comerciais que são normalmente utilizados na área da Integração de Processos.

No Capítulo 3 é apresentado o procedimento algorítmico-heurístico Diagrama de Fontes de Água aplicado aos sistemas uni e multicontaminantes.

No Capítulo 4 é apresentado o *software* desenvolvido, o MINEA 2.0. As etapas de aplicação do *software* são mostradas por meio de um exemplo e, em seguida, são aplicados alguns exemplos encontrados na literatura.

Por fim, no último capítulo estão reunidas as conclusões referentes à aplicação do MINEA 2.0 e sugestões futuras.

Capítulo 2 – Revisão Bibliográfica

Tendo em vista que o presente trabalho apresenta um novo *software* para a aplicação de problemas que envolvem a minimização de efluentes aquosos, o foco desse capítulo é voltado para os *software*s já desenvolvidos na mesma área e áreas afins. Todavia, o entendimento do funcionamento de vários desses *software*s depende de alguns conceitos básicos que fazem parte da Integração de Processos (mais especificamente, da Integração Mássica). Assim, conceitos fundamentais concernentes à Integração Mássica serão apresentados na primeira seção desse capítulo. Na seção seguinte encontra-se a revisão dos *software*s desenvolvidos até então.

A Integração de Processos é um ramo da Engenharia de Processos iniciada no final dos anos de 1970 e que se refere a certas atividades orientadas para um sistema com abordagens integradas para a análise, a síntese e o *retrofit* de plantas industriais (MANN, 1999).

Muitos autores adotam diferentes definições para a Integração de Processos. O essencial, entretanto, independente da definição adotada, é que o termo se refira a um conjunto de atividades sistemáticas e gerais, que sejam aplicadas aos processos produtivos, de forma a promover o uso mais eficiente da matéria e da energia, buscando reduzir os danos ao meio ambiente.

O conceito mais importante e que originalmente marca o nascimento da Integração de Processos ficou conhecido como *pinch* e advém dos primeiros trabalhos desenvolvidos na área de Integração Energética. O conceito foi posteriormente estendido para outras áreas usando diversas analogias, como por exemplo, o *Pinch* de Massa (gargalos termodinâmicos que excluem qualquer transferência de massa integrada entre as correntes ricas e as correntes pobres de um processo), desenvolvido por El-Halwagi e Manousiouthakis (1989); o *Pinch* de água, desenvolvido por Wang e Smith, em 1994; e o *Pinch* de Hidrogênio. O *pinch* é uma ferramenta que fornece informações cruciais a respeito da planta industrial (Gundensen, 2002).

Além do *pinch*, três conceitos normalmente associados à Integração de Processos são a análise, a síntese e a otimização de processos. A síntese consiste na geração das estruturas viáveis do projeto, na seleção dos equipamentos e da definição do fluxograma. Trata-se de um problema essencialmente combinatório, caracterizado por uma multiplicidade de soluções. A análise de processos é encarregada de identificar os elementos do sistema e a forma como eles interagem, seguida da previsão (aplicação de um modelo matemático) e do desempenho (análise econômica) do projeto (PERLINGEIRO, 2005).

Matematicamente, a síntese de processos pode ser descrita como a busca pela combinação ótima de unidades operacionais, as quais são conectadas em um arranjo ótimo e que operam nas condições de *otimalidade* obedecendo as especificações do projeto. (STEIMEL, 2014)

Uma vez que as estruturas viáveis foram identificadas, inicia-se a busca pela estrutura que melhor atende a um determinado objetivo. Todo projeto passará por um processo de otimização, ou seja, quais são as alterações que podem ser realizadas a fim de aumentar a eficiência e o desempenho da planta de processos. Essas mudanças podem ser feitas a partir da síntese das estruturas alternativas (otimização estrutural), seguida da simulação e a avaliação do processo e, por fim, a otimização da estrutura. Alternativamente, cada estrutura pode ser submetida a uma otimização paramétrica, alterando as condições operacionais dentro dessa estrutura (SMITH, 2005).

O problema a ser otimizado pode ser representado como um conjunto de equações. O objetivo da otimização é encontrar os valores das variáveis de processo que fornecem o melhor valor para o critério de otimização adotado (maximizar o rendimento, minimizar o custo de produção, maximizar a produtividade, minimizar a carga de efluentes etc), obedecendo a determinadas restrições do processo. Em termos técnicos, o objetivo da otimização é encontrar o melhor valor para a função objetivo obedecendo certas restrições.

Matematicamente, a formulação do problema de otimização possui a forma geral (ANTONIOU e LU, 2007):

$$\text{Maximize ou minimize } F = f(\mathbf{x}), \quad \mathbf{x} \in E^n$$
$$\text{Sujeito a: } a_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, p$$
$$c_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q$$

Onde:

$\mathrm{f}(\mathbf{x})$ é a função objetivo

$\mathbf{x}$ é o vetor dos parâmetros ou variáveis de decisão

a_i é o vetor das restrições de igualdade

c_j é o vetor das restrições de desigualdade

Atualmente, a grande maioria dos problemas de otimização são resolvidos utilizando a programação matemática, disciplina que engloba a teoria e a prática dos métodos numéricos para solucionar os problemas de programação linear, inteira, não linear, quadrática e dinâmica. Basicamente, as diferenças entre esses quatro ramos estão relacionadas à estrutura do problema de otimização e à natureza da função objetivo e das restrições.

Uma quantidade considerável de trabalhos foi publicada no campo da IP com os autores propondo diferentes formas de solução para um dado problema de síntese de processos. Coube, então, classificar as diferentes propostas de solução em função da ferramenta utilizada. Assim, três principais linhas podem ser observadas (GOMES, 2002):

- Sistemas Especialistas: métodos de síntese de processo e de integração de processos construídos com base no conhecimento acumulado de ideias já provadas. Entram nesse conjunto os métodos heurísticos, onde a sequência lógica da evolução do fluxograma segue regras geradas com base em experiência prévia.
- Procedimento Algorítmico: inclui-se nessa linha a tecnologia *pinch*, uma técnica de análise sistemática do comportamento de correntes de processos industriais baseada em fundamentos termodinâmicos.

- Programação Matemática: conjunto que engloba os cinco principais ramos da programação matemática mencionados anteriormente.

Apesar de os primeiros trabalhos relacionados à Integração de Processos datarem de 1970, somente nos anos de 1980, começaram a ser investigados os problemas relacionados à redução do consumo de água, denominados Problemas de Distribuição de Água/Efluentes (WAP – *Water/Wastewater Allocation Problems*).

Dado um conjunto de correntes de processos que utilizam água, os WAP referem-se a um problema no qual deseja-se encontrar a rede de interconexões das correntes aquosas envolvidas no processo de forma a reduzir o consumo de água primária global (SAVELSKI e BAGAJEWICZ, 2003). Esses problemas de distribuição de água tem sido amplamente formulados como problemas de minimização de consumo de água primária, nos quais o objetivo de minimizar o consumo é alcançado através do reuso e/ou reciclo envolvendo ou não processos de regeneração (FARIA e BAGAJEWICZ, 2010).

De forma geral, as estratégias para resolver WAP podem ser divididas em duas classes (BAGAJEWICZ, 2000):

(i) métodos baseados na programação matemática;

(ii) métodos de abordagem conceitual, ou seja, métodos gráficos, procedimentos algorítmicos ou aqueles baseados em heurísticas.

As duas classes apresentam vantagens e desvantagens. Os procedimentos que utilizam programação matemática são mais robustos, podendo fornecer soluções que encontram o ótimo global e são capazes de incorporar todos os tipos de restrições do problema. Além de minimização de água, a função objetivo pode apresentar outros fatores como custo ou número de conexões. Entretanto, os métodos de programação matemática não permitem a influência do projetista ao longo do cálculo e podem exigir grande esforço computacional dependendo da dimensão do problema. Por outro lado, os métodos de abordagem conceitual são muito limitados na resolução de problemas que envolvem muitas espécies

de contaminantes e não há garantias que o ótimo global foi atingido, mas normalmente são métodos simples, de fácil aplicação e que permitem a compreensão plena e imediata do projetista.

Os métodos baseados em programação matemática são fudamentados na otimização de uma superestrutura e seguem a formulação geral de um problema de otimização apresentada no Capítulo 1 dessa dissertação. O principal desafio da resolução dos WPA através da programação matemática é a presença de entidades não-lineares. Savelski e Bagajewicz (2000a) mostraram que problemas envolvendo um contaminante podem ser linearizados e apresentaram condições necessárias de *otimalidade* de utilização da água. Em 2003, os mesmos autores apresentaram as condições para os sistemas com múltiplos contaminantes. Diversos artigos usaram a programação matemática como forma de resolução para os problemas de minimização de efluentes aquosos. Bagajewicz (2000) apresenta uma revisão que compila as técnicas voltadas para alcançar as metas de consumo e a síntese de rede de água, com ênfase para as técnicas de programação matemática.

Um conceito fundamental da Integração Mássica é a Rede de Transferência de Massa (RTM). A RTM é uma rede que consiste de um ou mais trocadores de massa que, sinergeticamente, miminizam o consumo de água e a geração de efluentes (DUNN e EL-HALWAGI, 2003). O conceito de RTM foi primeiramente apresentado por El-Halwagi e Manousiouthakis (1989) como uma rede de transferência de massa, exequível em termos de custo, com o propósito de transferir espécies contaminantes de um conjunto de correntes ricas (o qual a massa deve ser retirada) para um conjunto de correntes pobres (o qual a massa deve ser adicionada). A partir das correntes, pode-se construir a Tabela de Oportunidades (ou Tabela de Intervalos de Composição – *Problem Table*) e, então, encontrar a metas de projeto. A metodologia de síntese proposta pelos autores teve como base os conceitos desenvolvidos para a Síntese de Rede de Trocadores de Calor e, portanto, analogamente à Integração Energética, os autores definiram os pontos de estrangulamento (*pinch points*) como gargalos termodinâmicos que excluem qualquer transferência de massa integrada entre as correntes ricas e as correntes pobres.

Wang e Smith (1994a) utilizaram o método gráfico para definir a meta de mínimo consumo de água. No método gráfico, os valores das concentrações das operações que usam água são inseridos em um gráfico de concentração *versus* carga mássica, criando um perfil de água. Quando as concentrações máximas de entrada e de saída das operações são plotadas em um gráfico concentração *versus* carga mássica, define-se o perfil limite de água, usado para definir uma fronteira entre a região das concentrações viáveis e a região das concentrações não-viáveis. A concentração de um perfil de água será considerada viável enquanto se mantiver abaixo do perfil limite de água. Sendo assim, para especificar o mínimo consumo de água primária considerando a opção de reuso, os perfis limites de água de todas as operações são plotadas no gráfico carga mássica *versus* concentração. Os perfis limites de água formarão a curva composta limite e, quando uma linha que representa a alimentação é inclinada até encostar na curva composta, o mínimo consumo é encontrado. O ponto onde a linha de alimentação de água encosta a curva composta limite é o ponto de *pinch*.

No campo da Integração Energética, a Análise *Pinch* é uma metodologia para o projeto conceitual de sistemas que envolvem a utilização de energia. A metodologia permite a maximização da utilização da energia dentro do processo, a minimização do uso das utilidades da planta de processos e efetua a análise dos *tradeoffs* entre os custos e a execução do projeto de minimização de energia (PERRY, 2008). Analogamente, a Análise *Pinch* na área da Integração Mássica, permite a maximização de recuperação de massa e a minimização do agente de transferência de massa. Os métodos para resolver WAP reduzem eficientemente o consumo global de água primária em uma planta de processos. A Análise *Pinch* é um método simples e apresenta excelentes resultados, apesar de não garantir o ótimo global.

Nos problemas que adotam os métodos de abordagem conceitual, a síntese de redes de água (ou minimização de água) é um caso especial dos problemas de síntese de redes de transferência de massa. Assim como outras aplicações típicas que utilizam a Análise *Pinch* (como, por exemplo, a síntese de redes de

transferência de massa ou de calor), a síntese de redes de água é subdividida em dois estágios: cálculos das metas de consumo mínimo e a síntese da rede (FOO, 2009).

Uma rede de água é uma coleção de processos que utilizam água (*water-using process*) e de operações de tratamento de efluentes. Elementos inerentes das redes de água são as fontes de água primária e os pontos de descarte dos efluentes. Os processos que utilizam água são classificados em dois grupos: (i) operações de transferência de massa (apresentam carga fixa); (ii) unidades que não envolvem transferência de massa ou unidades fonte e sumidouro (operações que apresentam vazão fixa). O primeiro grupo é caracterizado pela taxa de massa constante de contaminantes que são transferidos para as correntes aquosas e pelos valores máximos de concentração dos contaminantes. No segundo grupo, uma operação será caracterizada como um sumidouro de água se o processo somente consome água (por exemplo, água de resfriamento). As fontes de água do segundo grupo possuem valores fixos de vazão e de concentração dos contaminantes das correntes aquosas; os sumidouros também apresentam vazão fixa, mas a concentração dos contaminantes é variável e apresenta valores limites (JEZOWSKI, 2010).

2.1. *Software*s Desenvolvidos

Essa seção destina-se a apresentar alguns *software*s desenvolvidos para efetuar a solução de problemas na área de Integração de Processos, modelagem e otimização, com ênfase nos *software*s que sintetizam a Rede de Transferência de Massa, desenvolvidos no meio acadêmico ou no meio comercial, voltados para os campos da educação, pesquisa e indústria.

Dos trabalhos pesquisados, observou-se que a maioria dos autores que adotam a programação matemática como ferramenta para a minimização de efluentes aquosos, utilizam algum dos pacotes comerciais citados adiante. Quanto aos trabalhos pertencentes ao grupo da Análise *Pinch*, a produção científica na área de Integração Energética é bem mais numerosa que na Integração Mássica. O

intuito dessa revisão é determinar a tendência dos *software*s voltados para a área da Integração Mássica e identificar lacunas de funcionalidades.

A maioria dos *software*s encontrados são híbridos, isto é, realizam em uma primeira etapa a Análise *Pinch* para encontrar as metas de mínimo e, na segunda etapa, utilizam a programação matemática para encontrar a rede de transferência de massa através de algum solver comercial.

Existem diversos *software*s de simulação, disponíveis comercialmente, voltados para o cálculo dos parâmetros e das condições do processo envolvendo balanço de massa e energia, como o Aspen HYSYS, o CHEMCAD, o PRO/II, o UniSim Design entre outros. A maioria dos simuladores possui funções para estimar as propriedades físicas e de equilíbrio líquido-vapor, dimensiona equipamentos, possui biblioteca com as operações unitárias mais usadas e os modelos termodinâmicos, otimiza o fluxograma de produção e diversas outras características gerais envolvidas na simulação de processos químicos e industriais.

As ferramentas de simulação são frequentemente usadas para analisar projetos de economia de energia e sustentabilidade. Os simuladores auxiliam no desenvolvimento de modelos matemáticos e modelos completos de massa e energia baseados nos parâmetros de projeto. Essas ferramentas desempenham um papel importante na tomada de decisão que envolve aspectos técnicos e econômicos de projetos em execução ou em elaboração (LAM *et al*., 2011).

No campo da otimização existem diversos *software*s disponíveis comercialmente para solução de problemas de programação matemática, sejam esses problemas inseridos na otimização de processos químicos ou em qualquer outra área de aplicação da otimização.

Os *software*s de planilhas eletrônicas, como o Solver do Excel e o What's Best! da LINDO *Systems*, são capazes de resolver problemas de programação de dezenas de varáveis de decisão e restrições. O Solver do Excel é bastante popular e facilmente obtido. O *software* What's Best! é uma extensão (*add-in*)

para ser instalada dentro do Excel e é utilizado para a resolução de problemas de programação linear, não linear e inteira. O What's Best! possui ainda compatibilidade com o VBA (*Visual Basic for Applications*), possibilitando o usuário desenvolver macros e códigos de programação.

Também disponíveis no mercado, existem *software*s que usam linguagens de programação de alto nível para solução de problemas de otimização de larga escala ou problemas de programação matemática complexos. Um dos exemplos mais populares desse tipo de ferramenta é o GAMS (*General Algebraic Modelling System*), desenvolvido pela GAMS *Development Corporation*. O GAMS é um sistema de modelagem de alto nível, desenvolvido para resolver problemas de programação linear, não linear e mista inteira, possuindo interface para se conectar com outros pacotes de otimização.

Entre outros exemplos de pacote de otimização, pode-se citar os *software*s Lingo e LINDO (ambos da LINDO Systems), o CPLEX, o XPRESS (*Dash Optimization Ltd*), o OSL (*Optimization Subroutine Libary*) da IMB, o AMPL (*A Mathematical Programming Language*) da Bell e o AIMMS (*Advanced Integrated Multidimensional Modeling Software*) da *Paragon Decision Technology* (Belfiore e Fávero, 2013).

SPRINT, STAR, WORK, WATER, SuperTarget® e HEXTRAN são exemplos de *software*s disponíveis comercialmente e especificamente voltados para a área de Integração de Processos. Excetuando-se o *software* WATER (desenvolvido pela CEAS - *School of Chemical Engineering and Analytical Science* – antiga UMIST - *University of Manchester Institute of Science and Technology*), o SPRINT, o STAR, o WORK, o SuperTarget® e o HEXTRAN são *software*s voltados exclusivamente para a integração energética.

O WATER é voltado para o projeto dos sistemas que utilizam água nas indústrias de processo. O *software* identifica as metas de mínimo consumo de água e encontra, utilizando a programação matemática, a Rede de Transferência de Massa para os casos de máximo reuso, regeneração com reciclo, regeneração com reuso e múltiplas fontes de água externa. Possui uma função que encontra

automaticamente a rede de tratamento de efluentes tendo como critério de decisão a minimização do custo. O *software* pode ser aplicado para sistemas contendo um ou múltiplos contaminantes. O WATER não possui uma versão de teste comercial e seu uso está restrito a membros do *Center for Process Integration*, da Universidade de Manchester.

O Aspen Water™, da AspenTech®, utiliza a otimização matemática para a redução da geração de efluentes e dos custos operacionais. O Aspen Water™ é uma ferramenta para a síntese de rede dos processos que utilizam água, identificando os pontos de otimização da planta, fornecendo ao usuário diferentes opções de minimização da função objetivo como o custo da água, custo de conexões, custo com a regeneração e os custos referentes ao descarte dos efluentes. A principal vantagem do *software* é o banco de dados das operações que utilizam água e das operações envolvidas nas etapas de tratamento, o que possibilita a seleção ou a eliminação de operações do fluxograma e como isso afeta o processo em termos de custos.

O *software* MATLAB™ da MathWorks®, é uma linguagem de alto nível voltada para a computação numérica, a visualização e o desenvolvimento de aplicativos. O *software* possui uma variedade de métodos numéricos que permitem a análise de dados, o desenvolvimento de algoritmos e a criação de modelos. Possui interface que se comunica com outras linguagens como C, Java, .NET, and Microsoft® Excel®.

O *software* AquoMin foi apresentado em 1999 por Castro *et al* e, segundo os autores, trata-se de um *software* de interface amigável, desenvolvido no Visual Basic® da Microsoft. O *software* é voltado para a construção de redes de transferência de massa para problemas com um contaminante e baseia-se na tecnologia *pinch* e na programação matemática. O algoritmo do *software* para calcular o mínimo consumo de água fresca e a quantidade de água regenerada é o mesmo algoritmo proposto por Castro et al. (1999) e baseia-se na construção de um diagrama de intervalos de concentração. Em 2004, Relvas *et al.* atualizaram o AquoMin e criaram uma interface com o GAMS para resolver um problema de programação não-linear e determinar os parâmetros auxiliares

utilizados na formulação do problema, considerando a opção de regeneração com reuso. Os dados de entrada incluíam os dados limites do problema e a meta de mínimo obtida pelo diagrama de fontes de água, e os dados de saída do GAMS continham as informações da rede de transferência de massa. O programa híbrido ddsenvolvido pelos autores se mostrou eficaz na solução de problemas de minimização de efluentes considerando a regeneração com reuso.

No ano seguinte (2005), *Relvas et al.* apresentaram uma versão do *software* que incluía um novo módulo com opção para processos de regeneração com reciclo. O novo módulo possibilita encontrar a rede de transferência de massa diretamente a partir do diagrama de intervalos de concentração. O resultado obtido pelos autores comprova a eficiência do *software* em resolver problemas de mínimo consumo de água primária considerando a opção de regeneração com reciclo. Entretanto, o AquaMin é limitado a problemas contendo um contaminante. As demais limitações incluem o número de operações, número de fontes externas e o número de regeneradores.

Puigjaner *et al*. (2000) exploraram o campo dos processos em batelada e desenvolveram um *software* que possui um módulo que efetua redução de energia e um outro que calcula a demanda de água primária e o custo envolvido no sistema de reuso. As correntes aquosas nos processos em batelada não permanecem constante ao longo do tempo e, por essa razão, a otimização matemática deve incluir, além das restrições referentes às concentrações, as restrições referentes ao tempo. O processo de otimização gera uma rede que inclui o melhor aproveitamento dos tanques de estocagem de água usada. O *software* foi desenvolvido para ser utilizado em ambiente Windows® e o solver adotado para efetuar a programação matemática foi o GAMS/CONOPT. Os autores aplicaram exemplos da indústria de alimentos e obtiveram redução no consumo de água e de energia.

Prakotpol e Srinophakun (2004) desenvolveram um modelo baseado em algoritmos genéticos para resolver o problema de minimização de efluentes industriais. O modelo de programação não-linear inteira mista proposto pelos autores foi implementado em ambiente MATLAB, além da criação de uma Guia

de Interface do Usuário para facilitar a inclusão dos dados de entrada. Como base de comparação, os autores executaram os mesmos exemplos no LINGO. No caso de um contaminante para as opções de reuso e regeneração com reciclo, o *software* desenvolvido encontrou o mesmo valor de mínimo consumo de água que o valor encontrado utilizando o Lingo, entretanto a rede encontrada não foi a mesma. A justificativa para essa diferença reside no módulo de custo presente no LINGO. Nos exemplos envolvendo múltiplos contaminantes, os autores encontraram resultados melhores, embora sutis, em relação aos resultados encontrados utilizando o solver Lingo. Provavelmente, os resultados encontrados para os problemas multicontaminantes devem-se à habilidade dos operadores do algoritmo genético em "driblar" os ótimos locais.

Ullmer et al. (2005) desenvolveram o *software* WADO™, o qual apresenta duas etapas de execução: a primeira, baseada em regras heurísticas, determina a superestrutura do problema; a segunda etapa utiliza otimização matemática para efetuar o problema de programação não-linear inteira mista e encontrar o menor custo da rede. O WADO™ foi construído utilizando o *software* G2™ em conjunto com os *solvers* XPRESSMP E LSGRG2. A interface gráfica foi realizada em Java. O *software* desenvolvido encontrou uma redução de 56% de água fresca, utilizando o custo da água como função objetivo.

Teles *et al*. (2007) criaram o OptWatNet, um *software* de síntese de rede de água de sistemas multicontaminantes. O problema de minimização do uso de água primária pode ser formulado como um problema de programação não-linear, partindo de uma superestrutura que incorpora todas as estruturas alternativas. São disponibilizadas quatro estratégias diferentes (sendo duas de programação linear) para inicializar o problema geral de programação não-linear. O *software* foi desenvolvido para ser utilizado em ambiente Windows®, com comunicação com o GAMS para resolver o problema de programação não-linear e gera a visualização da rede ótima encontrada. Os autores encontraram que, os módulos que efetuam uma programação mais complexa apresentaram resultados melhores que os resultados encontrados pelos módulos que usam programação linear.

Desenvolvido por Santos (2007), o MINEA é um *software* baseado em planilha eletrônica voltado para a síntese de redes de transferência de massa que possui como base de cálculo o algoritmo Diagrama de Fontes de Água proposto por Gomes (2002). Para o caso de processos que apresentam múltiplos contaminantes, a escolha da operação de referência e o ajuste das concentrações são baseados no cálculo de um fator restritivo. O programa desenvolvido pelo autor pode ser aplicado às situações de máximo reuso, proibição de reuso e perda de vazão. Para o caso de um contaminante, o MINEA apresenta boa concordância de resultados dos valores de mínimo consumo encontrado pelos autores do problema original, validando o algoritmo computacional; entretanto, em casos de problemas com múltiplos contaminantes, o programa pode levar a resultados diferentes em relação àqueles encontrados pelos autores do problema original, na condição de máximo reuso.

O *software* RCNet (NG *et al*., 2014) é baseado em planilha eletrônica e voltado para a síntese de Rede de Transferência de Massa. O desenvolvimento do RCNet é baseado na ferramenta de Análise *Pinch* para determinar as metas de consumo mínimo e utiliza as técnicas de programação matemática para gerar a Rede de Transferência. O *software* foi desenvolvido para a determinar a Rede de Transferência de Massa de sistemas aquosos uni ou multicontaminantes, mas é extensível para sistemas de recuperação de hidrogênio. Apesar de a metodologia desenvolvida apresentar bons resultados, o *software* precisa do Excel® e do suplemento *What's Best!* para realizar os dois estágios da síntese de rede; ademais a apresentação da rede gerada é na forma matricial fonte-sumidouro.

Por fim, conclui-se que a maioria dos *software*s comerciais possuem licença de alto de custo, exigem treinamento do usuário e as versões de teste são limitadas a somente algumas funções e bibliotecas. A maioria dos trabalhos encontrados apresentam *software*s locais e baseados em planilhas eletrônicas, o que justifica a demanda por *software*s mais robustos que apliquem a Análise *Pinch* combinada à otimização matemática. Nenhum dos *software*s encontrados é

capaz de identificar violações das condições operacionais na Rede de Transferência de Massa.

Capítulo 3 – Diagrama de Fontes de Água

Esse capítulo destina-se a apresentar o algoritmo Diagrama de Fontes de Água (DFA) desenvolvido por Gomes (2002). A primeira parte apresenta a metodologia aplicada ao caso de sistemas com a presença de um contaminante, seus passos fundamentais e as regras que possibilitam a construção do diagrama com a especificação do mínimo consumo de água e a síntese da Rede de Transferência de Massa (RTM). Um problema exemplo é aplicado a fim de demonstrar a sequência do algoritmo. Na segunda parte é apresentada a metodologia aplicada ao caso de sistemas multicontaminantes desenvolvida por Gomes *et al*. (2013), com a atualização das regras de escolhas do contaminante e da operação de referência. Um exemplo de dois contaminantes é usado para mostrar a sequência dos passos do algoritmo.

O algoritmo Diagrama de Fontes de Água (DFA) apresentado por Gomes (2002) é uma extensão da metodologia apresentada por Castro *et al*. (1999) e é aplicado a problemas de um e múltiplos contaminantes. O algoritmo ainda contempla problemas com múltiplas fontes de água, regeneração com reuso, regeneração com reciclo, sistemas com perdas referentes ao processo e problemas de restrição de vazão.

O DFA possui abordagem conceitual baseado nos métodos de Análise *Pinch*, capaz de efetuar simultaneamente a especificação de metas de consumo e a síntese da Rede de Transferência de Massa (RTM), além de localizar o *pinch* global.

De forma geral, um equipamento de transferência de massa é qualquer unidade de transferência de massa, de contato direto, contracorrente, que utiliza um agente extrator e que pode ser representado, esquematicamente, conforme a Figura 3.1. Exemplos de operações que transferência de massa incluem absorção, adsorção, troca iônica, extração por solvente e destilação, enquanto agentes extratores incluem solventes, adsorventes e resinas de troca iônicas. (MANN, 1999).

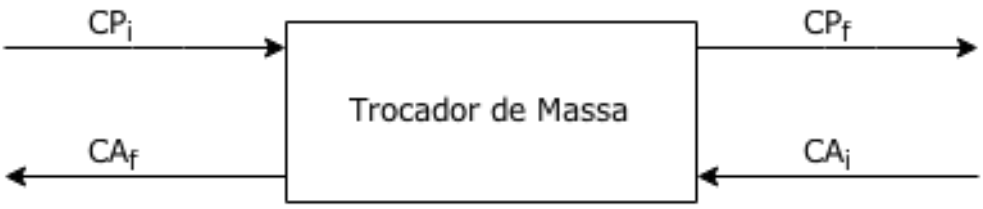

Figura 3.1 - Equipamento de Transferência de Massa (ETM). CP_i é a corrente de processo que entra no ETM; CP_f é a corrente de processo que sai do ETM; Ca_i é a corrente do agente extrator que entra no ETM; CA_f é a corrente do agente extrator que sai do ETM

Assim, considerando um conjunto de correntes de processo no qual há a necessidade de reduzir a carga de contaminantes através de um agente extrator, para aplicar o algoritmo, os únicos dados de entrada são as concentrações de entrada e saída de cada contaminante em cada operação e as vazões das operações. O procedimento constitui de quatro passos gerais, de três regras heurísticas e de uma equação de balanço de massa que rege todo o algoritmo:

$$\Delta m_k = G_k\left(C_{P,Ik} - C_{P,Fk}\right) = f_k(C_{fk} - C_{ik}) \qquad (3.1)$$

onde:

Δm_k, em g/h, é a taxa de massa de contaminante a ser transferida entre as correntes de processo e a corrente do agente extrator;

G_k, em t/h, vazão da corrente de processo;

$C_{P,Ik}$, em ppm, concentração inicial da corrente de processo;

$C_{P,Fk}$, em ppm, concentração final da corrente de processo;

f_k, em t/h, vazão da corrente do agente extrator;

C_{fk}, em ppm, concentração final da corrente do agente extrator;

C_{ik}, em ppm, concentração inicial da corrente do agente extrator.

Analisando a Equação (3.1), conclui-se que quando se eleva ao máximo o valor de C_{fk}, menor será a vazão necessária da corrente do agente extrator.

Considerando que o agente extrator é água, se $C_{ik} = 0$ (água primária) a vazão também é reduzida. Quando existe a possibilidade de reusar o efluente de uma operação como afluente para outra operação, a quantidade de água primária utilizada no processo é minimizada, atingindo o mínimo quando o reuso for máximo.

A corrente de água de um efluente será reusada em uma determinada operação se a sua concentração for menor que a concentração máxima de entrada da operação. Nos casos de máximo reuso, os dados limites de cada operação *k* ($f_k, C_{fk,max}, C_{ik,max}$) e da concentração da fonte externa de água (C_{fea}) são os únicos dados necessários para o desenvolvimento do algoritmo DFA. Quando o problema apresenta múltiplas fontes externas de água ou regeneração, ocorrem pequenas alterações no algoritmo. No presente trabalho, serão apresentados dois exemplos para ilustrar a aplicação do DFA nos casos de opção de máximo reuso com a presença de um contaminante e de múltiplos contaminantes.

.

3.1. Presença de um contaminante

Para ilustrar a aplicação do Diagrama de Fontes de Água para o caso de máximo reuso, será exposto o problema exemplo apresentado por Wang e Smith (1994). Adota-se, para a execução desse exemplo, que existe disponível somente uma fonte de água externa à concentração de 0ppm.

Tabela 3.1 – Dados do problema exemplo (Wang e Smith, 1994)

Operação (k)	f_k(t/h)	Contaminante (i)	$C_{ik,max}$ (ppm)	$C_{fk,max}$ (ppm)
1	20	0	100	2
2	100	50	100	5
3	40	50	800	30
4	10	400	800	4

Como já mencionado, a Equação (3.1) governa o algoritmo. Assim, a partir dos valores limites cada operação, a carga de contaminante é encontrada por:

$$\Delta m_k = f_k(C_{fk} - C_{ik}) \qquad (3.2)$$

O DFA é montado sequencialmente, seguindo quatro passos principais e obedecendo três regras, expostos compactamente a seguir (Gomes (2002) apresenta o detalhamento de cada passo sequencial do DFA aplicado a diversas aplicações para reduzir a geração de efluentes líquidos).

Passo 1: Dividir o diagrama em intervalos de concentração.

Nesse passo, todas as concentrações são adicionadas ao diagrama: a concentração da fonte externa de água e os limites de concentração das operações.

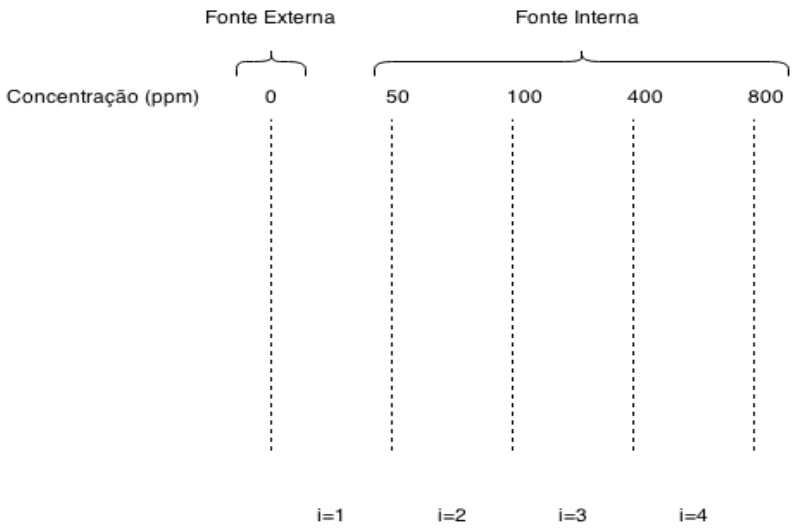

Figura 3.2 – Representação do Passo 1 do Diagrama de Fontes de Água

Passo 2: Representação das operações de transferência de massa para cada operação

As extremidades das setas representam os limites de concentração de cada operação. As vazões limites de cada operação também são adicionadas ao diagrama.

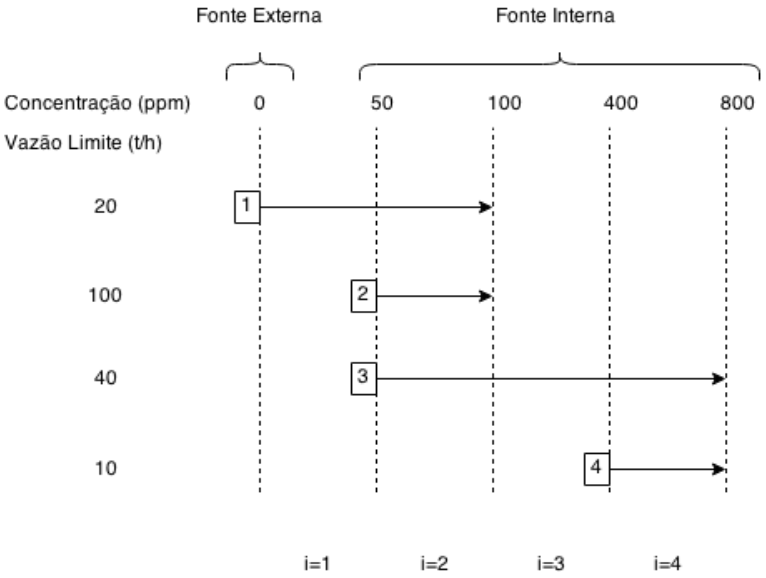

Figura 3.3 – Representação das operações no Diagrama de Fontes de Água (Passo 2)

Passo 3: Calcular a quantidade de massa transferida em cada operação *k* para cada intervalo *i*.

Nessa etapa, a Equação (3.2) é utilizada para cada intervalo do diagrama, ganhando a forma:

$$\Delta m_{ki} = f_k(C_{fi} - C_{ii}) \qquad (3.3)$$

Onde:

C_{ii} é a concentração inicial do intervalo;

C_{fi} é a concentração final do intervalo;

f_k é a vazão mássica limite da operação k, k = 1, ..., N_{op} e i = 1, ..., N_{int}

O resultado de cada Δm_{ki} é exibido entre parênteses.

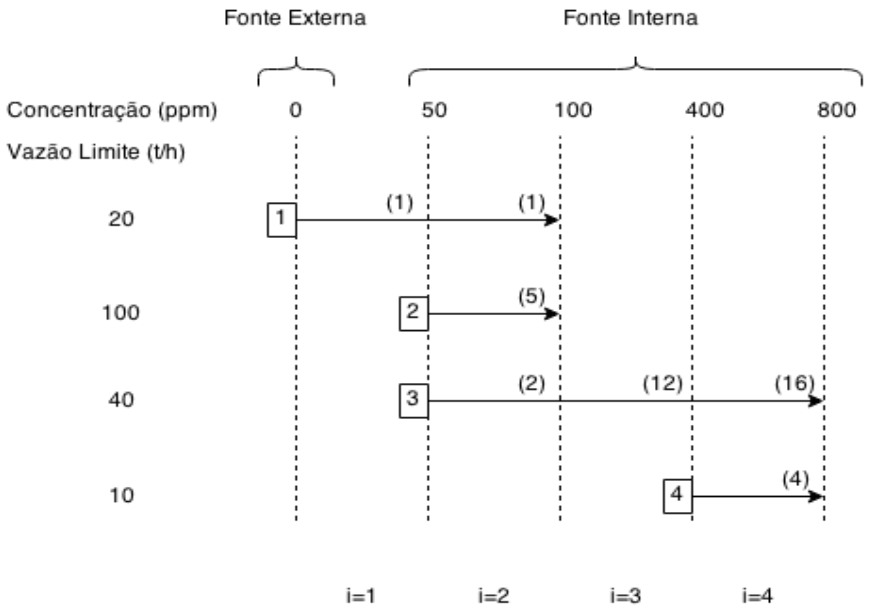

Figura 3.4 - Representação das operações no Diagrama de Fontes de Água (Passo 3)

Passo 4: Calcular a vazão necessária de água em cada operação para cada intervalo de concentração.

O procedimento para a construção de RTM inicia-se no intervalo de menor concentração. Para cada intervalo de concentração deve-se identificar todas as fontes (externas e internas) de água disponíveis no intervalo.

Para calcular a vazão mássica de cada intervalo, utiliza-se a Equação (3.3) e três regras devem ser satisfeitas:

- **Regra 1**: utilizar fontes externas somente quando não houver fontes internas disponíveis. Na disponibilidade de fonte interna, usar preferencialmente a oriunda da mesma operação
- **Regra 2**: para uma determinada operação, a fonte de água utilizada em um certo intervalo de concentração deve absorver a quantidade de massa a ser transferida no respectivo intervalo
- **Regra 3**: a prioridade de utilização das fontes internas é sempre da maior fonte de maior concentração para a de menor concentração.

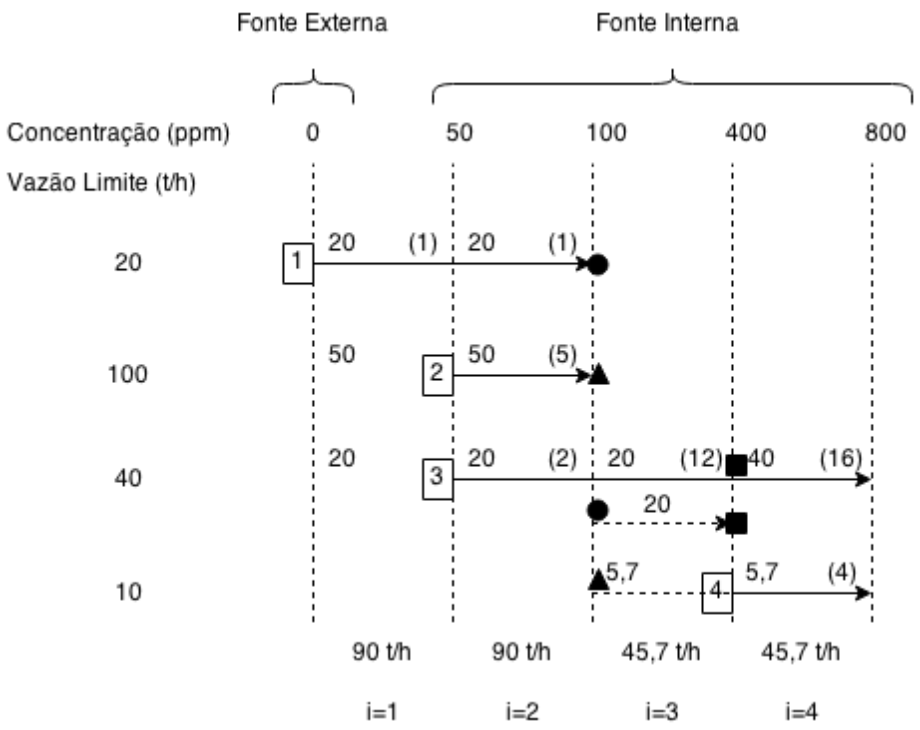

Figura 3.5 – Representação do Diagrama de Fontes de Água.

O consumo mínimo de água primária da fonte externa *p* é dado por:

$$f_p^e = \sum_{k=1}^{N_{op}} \sum_{i=1}^{N_{int}} f_{pki}^e \qquad (3.4)$$

O ponto *pinch* de concentração pode ser obtido identificando a concentração na qual a vazão total utilizada dentro do intervalo (soma das vazões utilizadas em cada operação, representada no DFA na última linha do diagrama) começa a cair.

A partir dos resultados obtidos no diagrama, a Rede de Transferência de Massa para o mínimo consumo de água primária é mostrada na Figura 3.6.

O *workflow* que representa a lógica de alocação das correntes aquosas e que sintetiza o algoritmo é apresentado na Figura 3.7.

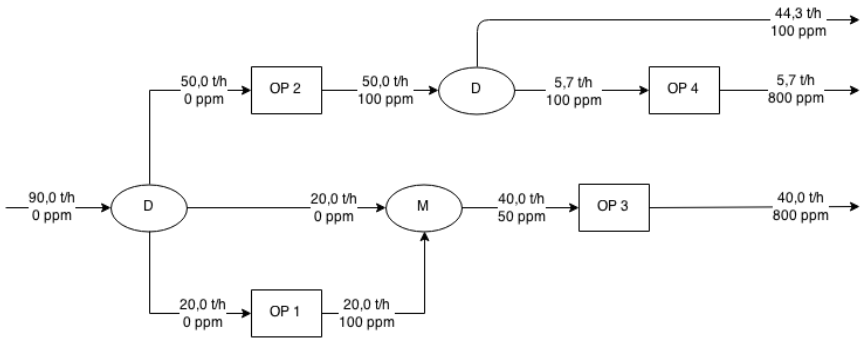

Figura 3.6 – Rede de Transferência de Massa para a opção de máximo reuso correspondente ao diagrama da Figura 3.5.

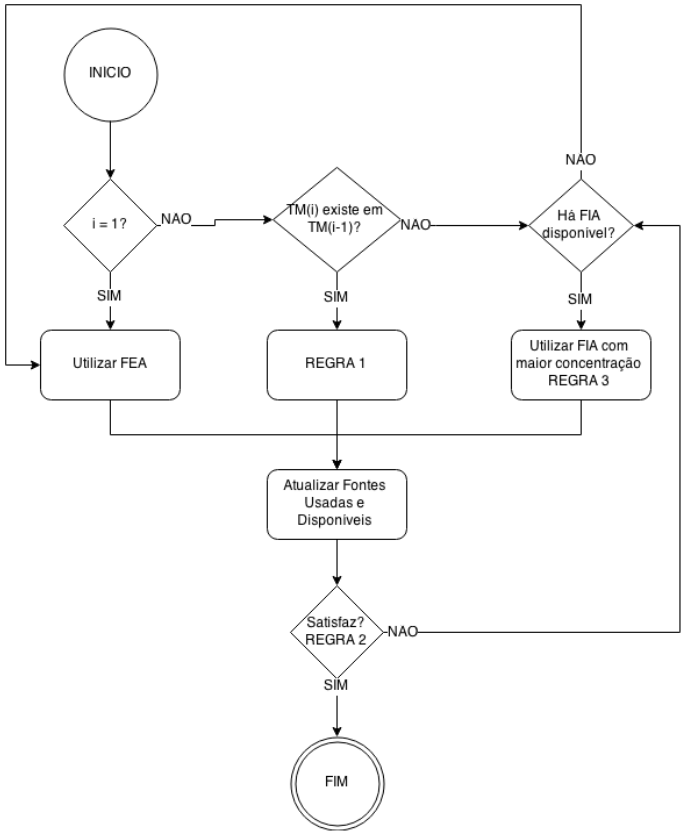

Figura 3.7 – *Workflow* do Diagrama de Fontes de Água de sistemas com um contaminante (FEA é a Fonte Externa de Água; FIA é a Fonte Interna de Água; TM é a transferência de massa para uma dada operação em um dado intervalo de concentração)

3.2. Presença de múltiplos contaminantes

Gomes et al. (2013) apresentaram uma extensão do algoritmo Diagrama de Fontes de Água para o caso de múltiplos contaminantes. O algoritmo proposto pelos autores é composto de sete etapas principais e sequenciais e efetua a síntese da rede de equipamentos de transferência de massa, fornecendo simultaneamente o consumo mínimo de água e a Rede de Transferência de Massa correspondente.

As operações de transferência de massa seguem a Equação (3.2), considerando cada contaminante *j*, em cada operação *k*, em seus limites máximos de concentrações de entrada e de saída. Assim:

$$\Delta m_{kj} = f_k\left(C^k_{fj,max} - C^k_{ij,max}\right) \qquad (3.5)$$

Em operações envolvendo múltiplos contaminantes, a transferência de massa ocorre simultaneamente. A relação linear de transferência dos contaminantes sugerida por Wang e Smith (1994) é adotada. Assim, em uma determinada operação *k*, a razão entre a taxa de massa transferida de qualquer contaminante *j* e a taxa de massa do contaminante de referência (ref) é uma constante (Kj,ref,k). Como a vazão é constante:

$$K_{j,ref,k} = \frac{\Delta C_{j,k}}{\Delta C_{ref,k}} = constante \qquad (3.6)$$

Para demonstrar o algoritmo e descrever cada uma das sete etapas, será adotado um problema exemplo apresentado por Wang e Smith (1994), mostrado na Tabela 3.2:

Tabela 3.2 – Dados de limites para o Problema Exemplo (Wang e Smith, 1994)

Operação (k)	$\boldsymbol{f_k}$(t/h)	Contaminante (i)	$\boldsymbol{C_{ik,max}}$ (ppm)	$\boldsymbol{C_{fk,max}}$ (ppm)	$\Delta\boldsymbol{m_k}$(kg/h)
1	40	A	0	100	4
		B	25	75	2

2	35	A	80	240	5.6
		B	30	90	2.1

Considera-se que a fonte externa de água disponível possui a concentração de 0ppm (0ppm de A e 0ppm de B)

Passo 1: Determinação do Contaminante de Referência e da Operação de Referência.

O contaminante de referência e a operação de referência são definidos a partir de um conjunto de regras:

- O contaminante de referência deve ser aquele que apresenta a menor concentração máxima de entrada em mais operações. Além disso, seguindo o critério proposto Salveski e Bagajewicz (2003), as concentrações de saída do contaminante de referência devem apresentar crescimento monotônico através das operações. Se essas regras indicarem mais de um contaminante, qualquer um deles pode ser escolhido.
- A operação de referência é aquela que necessitar da água de maior qualidade, ou seja, mais pura possível disponível a ser utilizada no processo. Além disso, na operação de referência, o contaminante de referência terá uma concentração máxima de entrada igual a sua concentração com a fonte externa de água mais limpa. No diagrama, a operação de referência será a primeira (número 1).

No exemplo, os dois contaminantes satisfazem as condições da primeira regra. O contaminante A é escolhido como referência, pois ele necessita de 0ppm na entrada da operação 1, que é operação definida como referência.

Passo 2: Ajuste das Concentrações de Entrada na Operação de Referência.

Neste segundo passo, as concentrações de entrada de todos os contaminantes na operação de referência são ajustadas para os valores da fonte de água externa. Como a concentração de entrada alterada, a respectiva concentração

de saída deve ser ajustada mantendo a transferência de massa da operação. A partir daí, a razão das diferenças de concentração ($K_{j,ref,k}$) utilizadas nesse procedimento são baseadas na transferência de massa na operação de referência (k = 1). Esses novos valores das concentrações de saída na operação 1 são aqueles que serão utilizados nos testes definidos no próximo passo (passo 3). No exemplo, como B possui uma concentração de entrada na operação de referência diferente de 0 ppm, esse valor deve ser assumido e a respectiva concentração de saída ajustada. Com A como o contaminante de referência, a razão na operação de referência ($K_{B,A,1}$) é:

$$K_{B,A,1} = \frac{\Delta C_{B,1}}{\Delta C_{A,1}} = \frac{75-25}{100-0} = 0{,}5 \qquad (3.7)$$

Assumindo $C_{iB}^{1} = 0ppm$, a concentração de saída respectiva de B é então calculada (50ppm B). Esses valores são exibidos entre colchetes na Tabela 3.3, sob a respectiva concentração de A.

Tabela 3.3 - Valores ajustados das Concentrações de B na Operação 1

Contaminante	**Concentração (ppm)**			
A	0	80	100	240
B (OP.1)	[0]			[50]

Passo 3: Identificação da Necessidade de Ajustes Baseada nas Operações Restantes.

A proposta desse passo é avaliar o comportamento dos demais contaminantes em todas as operações, exceto a de referência. O objetivo é garantir que as possibilidades de reuso do efluente da operação de referência estarão presentes em um DFA que teve a construção baseada exclusivamente no contaminante de referência.

Os demais contaminantes podem apresentar dois possíveis comportamentos:

(i) O contaminante *j* possui uma concentração de saída na operação de referência menor que todas as suas concentrações de entrada nas demais operações;

(ii) Existem uma ou mais operações onde a concentração de entrada do contaminante j não satisfaça a condição anterior.

No caso (i), não há necessidade de ajuste de concentração baseado no contaminante *j*. No caso (ii), um ajuste é necessário baseado na concentração do contaminante j, conforme descrito no próximo passo, para cada operação k onde a condição do caso (i) é falsa.

No exemplo, os dados das Tabelas 3.2 e 3.3 mostram que a concentração de entrada do contaminante B na operação 2 não satisfaz a condição do caso (i) e, portanto, deve ser realizado o ajuste das concentrações do contaminante de referência baseado nas concentrações de B na operação 2.

Passo 4: Ajuste de Concentrações do Contaminante de Referência para Oportunidades de Reuso.

Esse passo tem o objetivo de ajustar os valores das concentrações do contaminante de referência baseado nas concentrações de outros contaminantes, de forma a manter a compatibilidade das concentrações e então permitir a investigação de possibilidades de reuso para a corrente, saindo da operação de referência.

Na operação *k*, onde previamente foi identificado uma necessidade de ajuste baseado no contaminante *j*, o procedimento pode ser dividido em dois estágios:

(a) determinação da concentração correspondente do contaminante de referência à concentração de entrada de *j* na operação *k*, baseada na razão de transferência na operação de referência ($K_{j,ref,1}$);
(b) determinação da concentração correspondente do contaminante de referência à concentração de saída de j na operação k, agora baseada na razão da transferência de massa na operação k ($K_{j,ref,k}$).

Portanto, para o exemplo, a concentração de A na corrente da operação 1 correspondente à concentração de entrada de B na operação 2 ($C_{B,i|2}$ = 30ppm B) é computada, usando a razão $K_{B,A,1}$ da Equação 3.7 e as concentrações de entrada na operação 1. O valor obtido é, então, de 60ppm de A.

Ciente do valor ajustado da concentração de A correspondente à concentração de entrada de B na operação 2, o respectivo valor de saída é calculado. A vazão da transferência de massa usada é a da operação 2:

$$K_{B,A,2} = \frac{\Delta C_{B,2}}{\Delta C_{A,2}} = \frac{90-30}{240-80} = 0{,}375 \qquad (3.8)$$

Portanto, partindo dos valores de entrada de 60ppm de A e 30ppm de B, e utilizando o valor de saída de 90ppm de B e $K_{B,A,2}$, o valor obtido para a concentração de A correspondente é 220ppm. Esses resultados estão apresentados na Tabela 3.4, onde em colchetes está o resultado da concentração ajustada de A correspondente à concentração de entrada de B na operação 2 e em parênteses está a concentração de A correspondente à concentração de saída de B na operação 2.

Tabela 3.4 - Matriz das Concentrações Ajustadas

Contaminante	**Concentração (ppm)**					
A	0	[60]	80	100	(220)	240
B (OP.1)	**0**			**50**		
B (OP.2)		30			90	

Passo 5: Geração do Diagrama de Fontes de Água baseado no ajuste das concentrações do contaminante de referência.

Este passo consiste em executar o algoritmo Diagrama de Fontes de Água para um contaminante proposto por Gomes (2002) e apresentado primeiramente nesse capítulo.

Os dados do processo reescritos com suas concentrações ajustadas devido ao contaminante de referência são apresentados na Tabela 3.5

Tabela 3.5 - Problema Exemplo reescrito baseado no Contaminante de Referência

Operação (k)	f_k(t/h)	Contaminante (i)	$C_{ik,max}$ (ppm)	$C_{fk,max}$ (ppm)	Δm_k(kg/h)
1	40	A	0	100	4
2	35	A	60	220	5.6

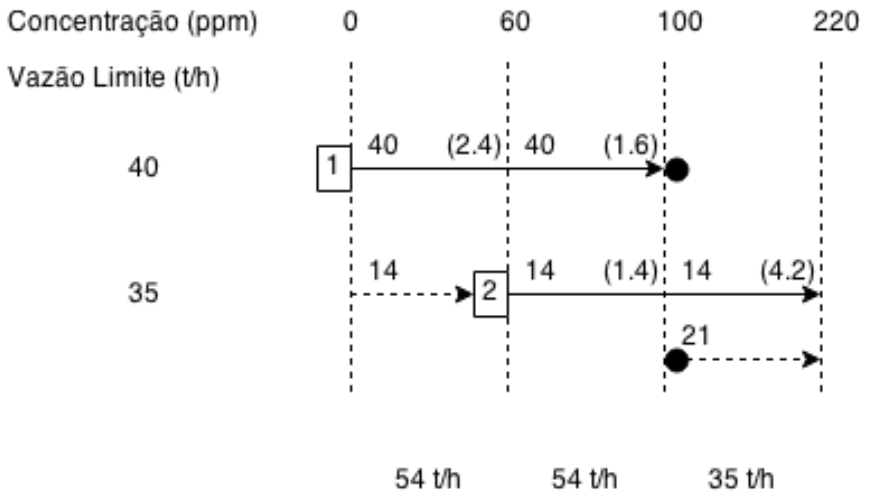

Figura 3.8 – Diagrama de Fontes de Água para o problema exemplo

Passo 6: Construção da Rede de Transferência de Massa e Inspeção das Outras Concentrações de Contaminantes.

A partir da estrutura do DFA final, é possível desenhar a rede de água correspondente. Além disso, a determinação da rede e o conhecimento das equações de balanço de massa permitem a determinação das concentrações de entrada e saída de todos os contaminantes em todas as operações. Em seguida, os limites originais das concentrações (no exemplo definido na Tabela 3.2)

podem ser verificados. Se alguma violação dos limites é percebida, ou seja, uma concentração maior que seu correspondente limite especificado nos dados originais da operação, a rede de água obtida precisa sofrer ajustes adicionais no próximo passo. De qualquer forma, a ausência de violações indica que a rede de água encontrada é o resultado final do algoritmo.

A rede encontrada (Figura 3.9) não exibe qualquer violação e, portanto, é o resultado final obtido pelo algoritmo estendido de DFA proposto para múltiplos contaminantes. O consumo mínimo é de 54 t/h de água pura calculado pelo DFA é igual ao valor obtido por Wang e Smith (1994) com o procedimento gráfico.

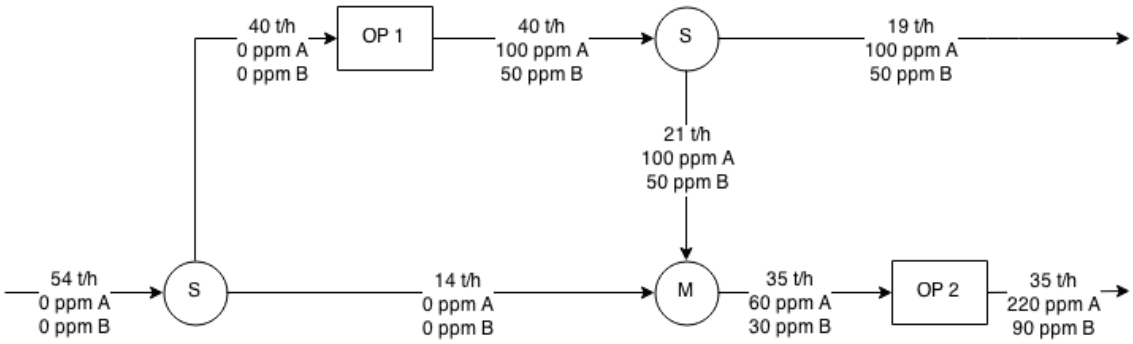

Figura 3.9 – Rede de Transferência de Massa corresponde ao diagrama da Figura 3.8.

Passo 7: Evolução da Rede para Remover as Violações

Quando há violações na Rede de Transferência de Massa, elas precisam ser removidas, já que os limites de concentração originais precisam ser respeitados pela rede final.

Algumas formas de eliminar as violações incluem:

(i) Aumentar o consumo de água externa;
(ii) Redirecionar as saídas de algum divisor de correntes anterior a operação onde a violação foi verificada. Essa opção é capaz de remover as violações sem aumentar o consumo de água externa, caso a operação onde ocorre a violação não seja alimentada por água primária.

Após a remoção das violações, os balanços de massa são realizados na rede para recalcular as novas concentrações. Em seguida, se alguma outra violação ainda permanecer, o procedimento de eliminação é repetido até que todas as violações tenham sido removidas.

O *workflow* do algoritmo de DFA para processos de múltiplos contaminantes é resumido na Figura 3.10.

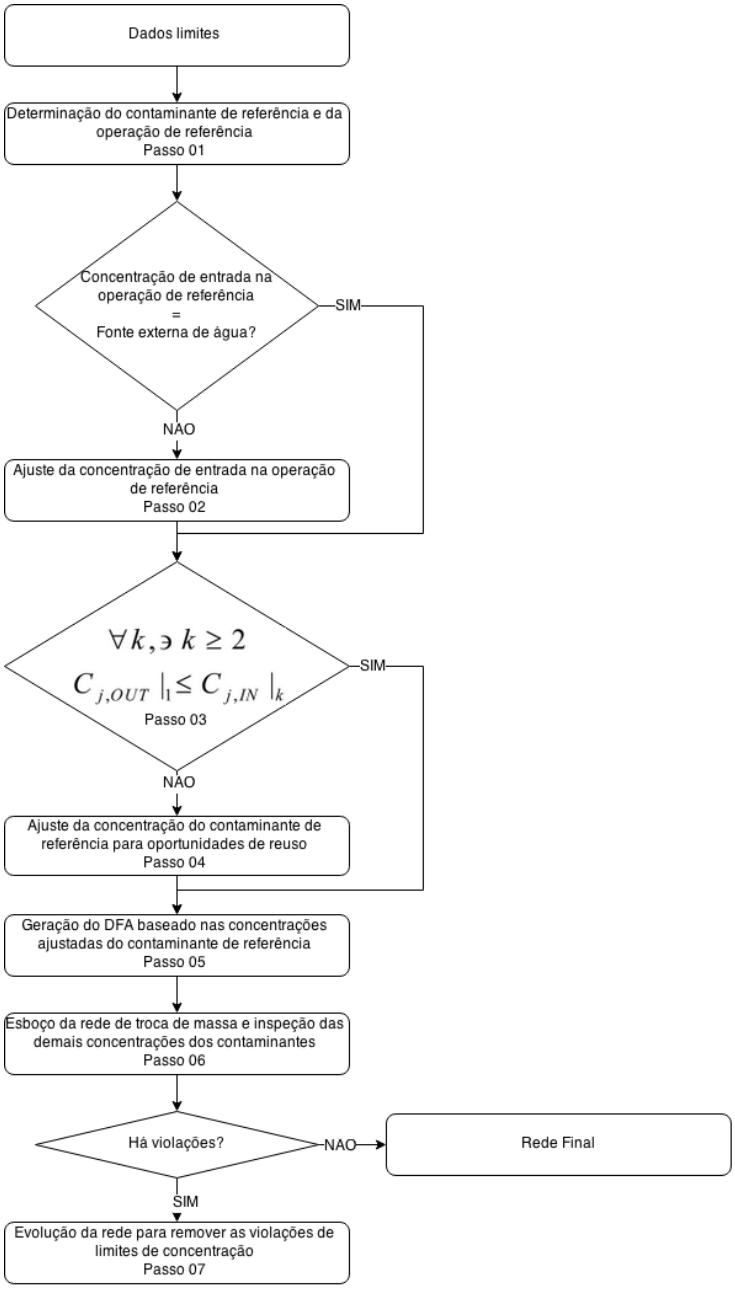

Figura 3.10 – *Workflow* do Diagrama de Fontes de Água para sistemas multicontaminantes (Gomes *et al*., 2013 – Adaptado)

Capítulo 4 - O *software* MINEA 2.0

O Diagrama de Fontes de Água apresentado no capítulo anterior é simples, rápido e fácil de ser aplicado manualmente. Entretanto, conforme o número de operações e contaminantes aumenta, maior a dimensão do problema e maior o tempo dispendido para a aplicação. A utilização de um *software* para a aplicação do DFA possibilita a análise de vários cenários em menor tempo porque, evidentemente, o processo computacional possibilita maior agilidade na resolução dos problemas.

Mais um fator motivacional advém do fato do DFA não garantir que o mínimo consumo de água primária das fontes externas seja realmente alcançado: o resultado obtido pelo DFA pode ser usado como estimativa inicial para aplicação em pacotes de otimização que utilizam programação matemática, como, por exemplo, o GAMS.

O MINEA, elaborado por Santos (2007), não era compatível com todas as versões de Excel®, apresentando instabilidades intermitentes durante sua execução.

No entanto, apesar da facilidade de uso das planilhas eletrônicas, elas possuem limitações quando se faz necessário elaborar cálculos mais complexos ou quando o volume de informações torna-se muito elevado. Com o intuito de minimizar essas limitações e permitir que ela seja atualizada e complementada no futuro, decidiu-se por elaborar a ferramenta em .NET, usando C# como linguagem de programação.

O C#, por ser orientado a objeto, permite que o programa seja elaborado de forma componentizada, ou seja, é possível que novos componentes de cálculo ou de exibição sejam adicionados ao programa original sem que seja necessário um grande esforço para adaptação do mesmo.

A estrutura dos componentes do *software* ficou da seguinte forma: (i) formulários de visualização; (ii) entidades; (iii) classes de cálculo. O primeiro refere-se ao conjunto de classes que representam cada uma das telas do programa. O segundo, refere-se às estruturas responsáveis pelo armazenamento das informações. Por fim, o terceiro refere-se à classe que realiza todos os cálculos para um contaminante ou para múltiplos contaminantes.

Além disso, cada etapa dos cálculos está presente em um método diferente dentro da classe de cálculos. Isso provê um desacoplamento entre as etapas dos cálculos, permitindo uma maior manutenibilidade e adaptabilidade do código-fonte.

Outro aspecto priorizado foi a elaboração de uma entidade responsável por armazenar o resultado do cálculo de cada etapa para cada intervalo. Dessa forma, caso seja necessário retrair algum dos passos dos cálculos em alterações futuras, também será possível, sem a necessidade de retrabalho.

Por fim, a linguagem C# possui uma grande variedade de componentes gráficos disponíveis que podem ser adicionados para melhor visualização da Rede de Transferência de Massa e do próprio Diagrama de Fontes de Água.

A atualização do *software* MINEA para linguagem C# recebeu o nome de MINEA 2.0 e todas as considerações para a construção do código para um contaminante e para múltiplos contaminantes foram baseadas, respectivamente, nos trabalhos de Gomes (2002) e Gomes *et al* (2013) apresentadas no Capítulo 3 dessa dissertação.

O MINEA 2.0 é uma aplicação *desktop*, construída para ambiente Windows®, que calcula o mínimo consumo de água para os casos de máximo reuso, múltiplas fontes de água e regeneração com reciclo para sistemas uni e multicontaminantes. Apesar de o MINEA 2.0 ainda não apresentar a capacidade de exibir visualmente a Rede de Transferência de Massa, todos os valores das correntes de água são calculados e apresentados textualmente.

O MINEA 2.0 possui interface amigável e intuitiva. Para apresentar o *software*, serão usados os dados limites do problema exemplo de múltiplos contaminantes apresentado por Wang e Smith (1994).

O programa foi construído na forma de navegação entre telas e se divide em duas partes principais: primeiro a entrada de dados e, ao final, a apresentação dos resultados. A entrada dos dados está dividida em três telas distintas, onde serão informados os dados das operações e das fontes externas de água, os dados dos contaminantes e as restrições de reuso, caso existam. Os resultados são apresentados inicialmente de forma gráfica, e há a possibilidade de gerar um relatório detalhando o resultado.

Na primeira tela, Figura 4.1, o usuário informa as operações com suas respectivas vazões e as fontes externas com suas respectivas concentrações. Caso alguma das fontes externas tenha vazão limitada, o usuário deve informar esse limite em “Vazão Disponível”. Se nada for informado sobre a disponibilidade, o *software* interpretará como vazão ilimitada. Em problemas envolvendo múltiplos contaminantes, as concentrações dos contaminantes presentes em cada fonte externa serão iguais à concentração da fonte externa informada pelo usuário.

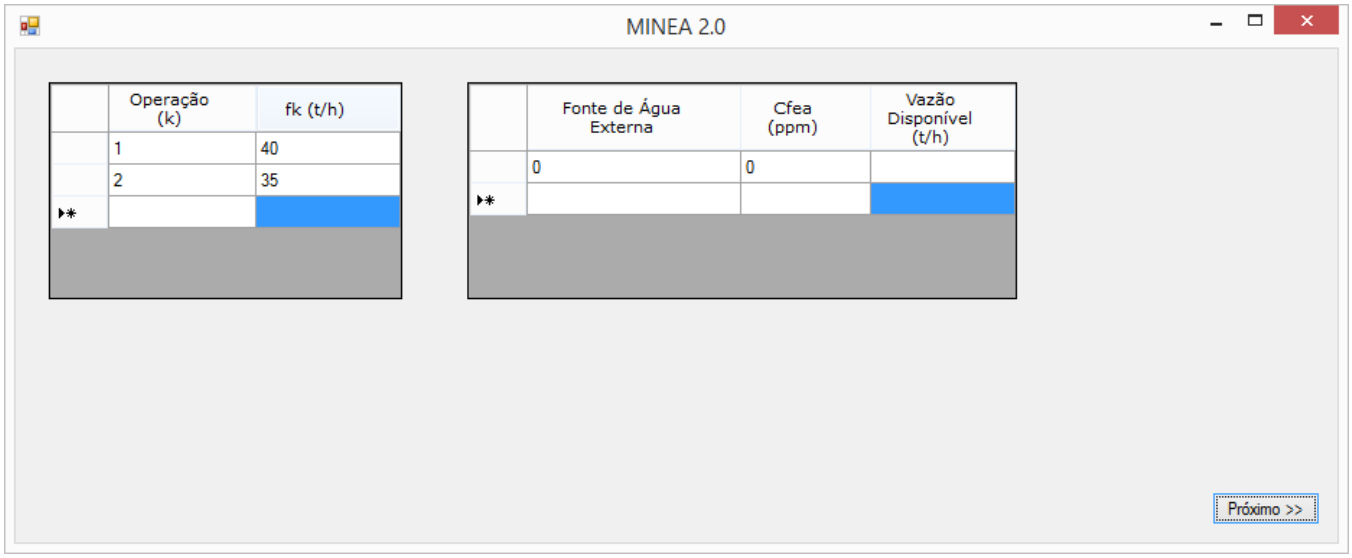

Figura 4.1 – Primeira tela de inserção de dados do MINEA 2.0

Na tela seguinte, Figura 4.2, são incluídos os dados sobre os as operações na Tabela de Oportunidades. O valor da taxa de transferência de massa é calculado automaticamente e expresso em g/h. O usuário deve informar qual é a operação de referência e o contaminante de referência em suas respectivas Caixas de Seleção. Os botões de Ajuda ao lado de cada Caixa de Seleção informam os

critérios de seleção (apresentados no Capítulo 4) para a escolha da operação e do contaminante de referência que o usuário deve se basear (Figuras 4.3 e 4.4).

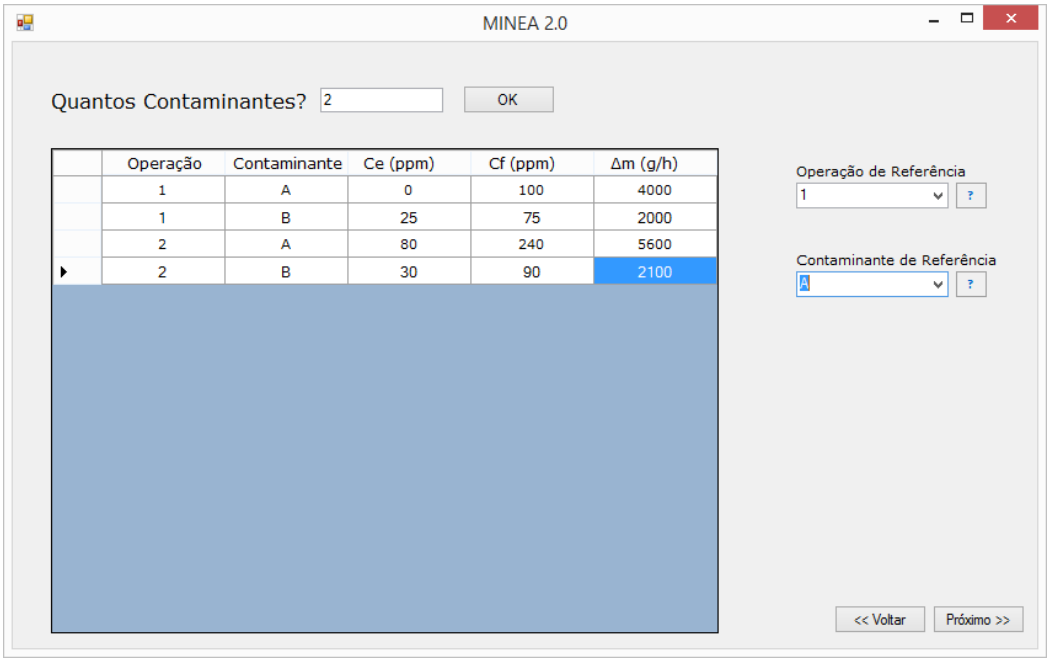

Figura 4.2 – Segunda tela de inserção de dados do MINEA 2.0

Na terceira tela da navegação (Figura 4.5), o usuário deve informar se existe alguma proibição de reuso do efluente de alguma operação. Por padrão, são permitidas todas as possibilidades de reuso entre as operações.

Por fim, a última tela (Figura 4.6) apresenta o Diagrama de Fontes de Água, gerado automaticamente assim que o usuário pressiona “Avançar” na tela anterior. As operações de transferência de massa são exibidas dentro do retângulo azul. Cada retângulo, em cada intervalo, contém todas as operações de transferência de massa realizadas em cada operação do processo. Na primeira linha da retângulo azul encontram-se as taxas de transferência para cada intervalo de concentração, calculadas utilizando a Equação 3.3. Nas linhas subadjacentes, são informadas as vazões e suas respectivas fontes de corrente de água. Assim, por exemplo, a linha “fea – 0:40” (no primeiro intervalo) informa que a fonte externa de água (fea) a 0ppm forneceu a vazão de 40t/h. No topo do Diagrama, são exibidos somatórios das vazões por intervalo de concentração. O consumo de água que cada fonte também é obtido diretamente do diagrama e corresponde ao somatório de cada vazão usada em cada operação.

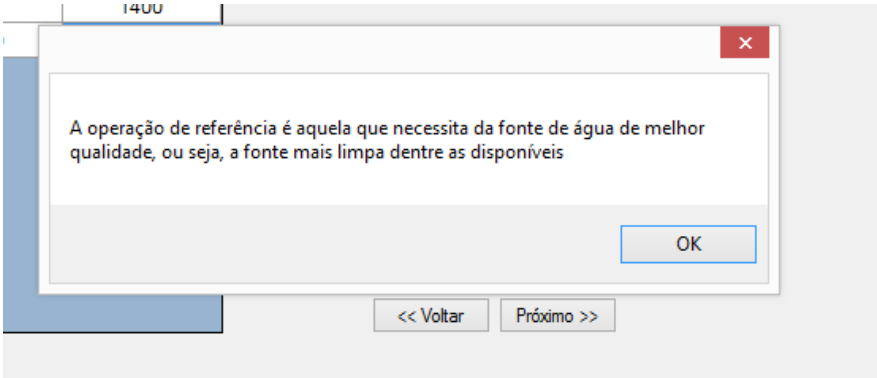

Figura 4.3 – Caixa de Diálogo do botão “Ajuda” pertencente à Caixa de Seleção da operação de referência

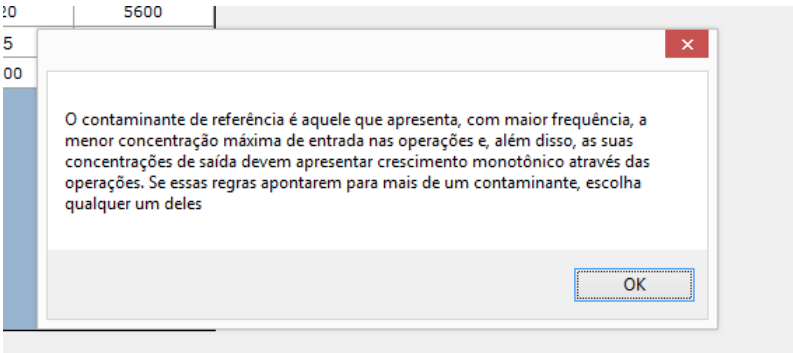

Figura 4.4 - Caixa de Diálogo do botão “Ajuda” pertencente à Caixa de Seleção do contaminante de referência

Ainda na última tela, caso o botão Gerar Relatório seja selecionado, será apresentado um relatório que resume as informações do problema e os resultados da Síntese de Rede de Transferência de Massa. As informações exibidas textualmente são (Figura 3.7):

1. Quantidade de operações presentes no problema e que foram fornecidas pelo usuário;
2. Quantidade de contaminantes presentes no problema e que foram fornecidas pelo usuário;
3. Quantidade de intervalos de concentração encontrada pelo MINEA 2.0; Vazão total da fonte externa usada e que representa o mínimo valor obtido pelo algoritmo Diagrama de Fontes de Água;
4. Vazão total por intervalo;

5. Para cada operação, a vazão total de fonte externa utilizada pela operação;
6. Para cada operação, a vazão de efluentes aquosos que foi reusada das fontes internas;
7. Para cada operação, a concentração de entrada e de saída de cada contaminante que foram encontradas pelo MINEA 2.0 (nos problemas de múltiplos contaminantes).

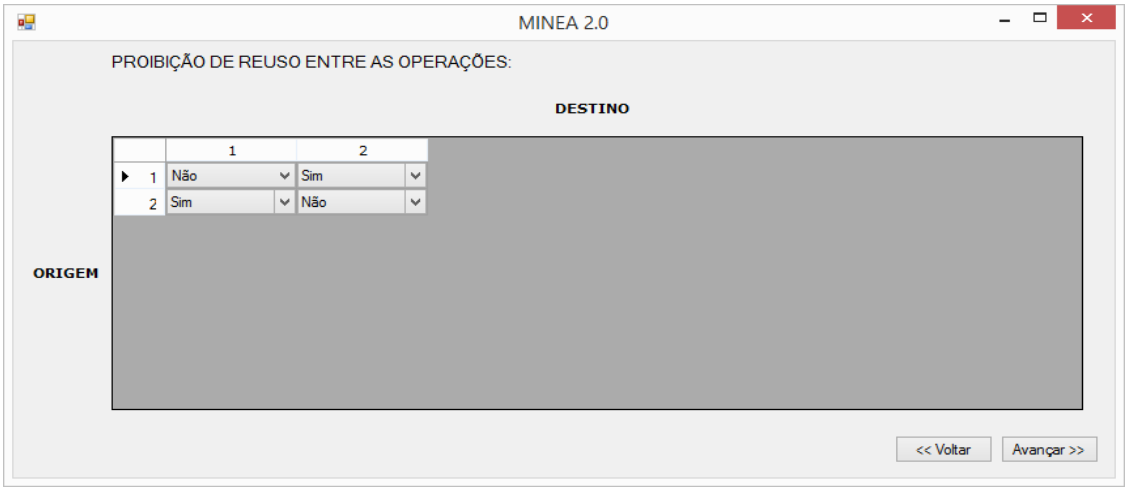

Figura 4.5 – Terceira tela do MINEA 2.0

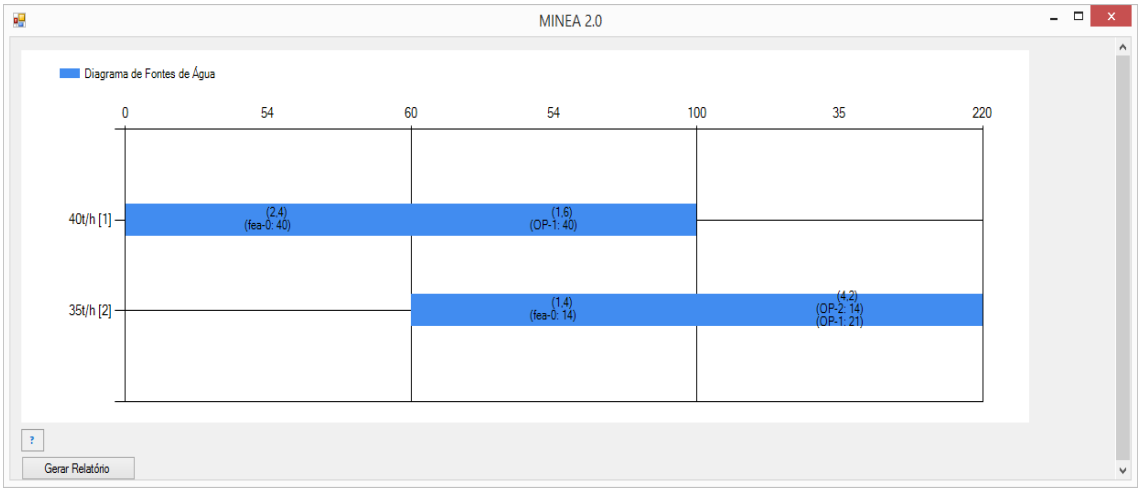

Figura 4.6 – Imagem do Diagrama de Fontes de Água criada pelo MINEA 2.0

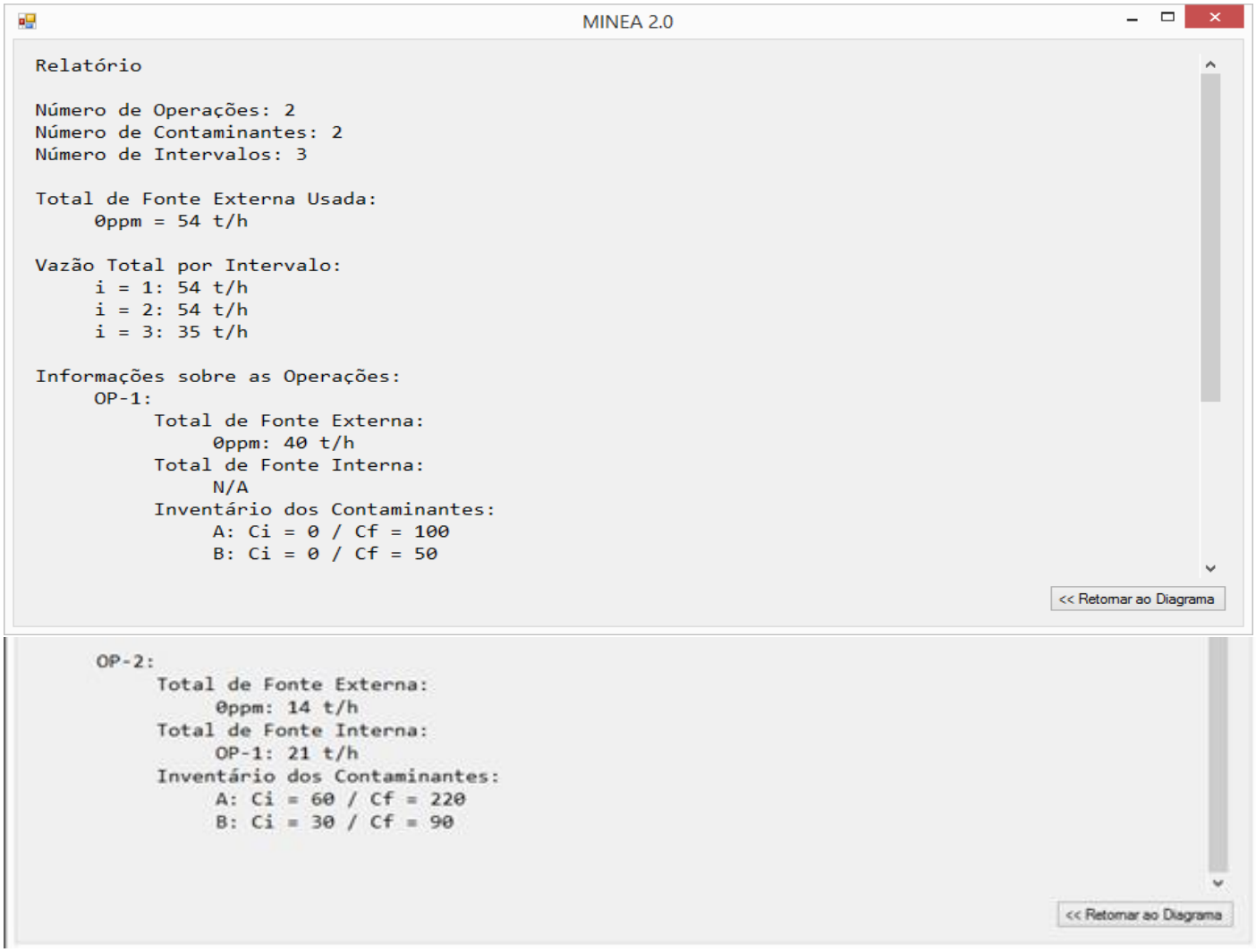

Figura 4.7 – Tela do “Relatório” com as informações sobre o diagrama e sobre as operações

A seguir serão apresentados alguns exemplos encontrados na literatura. Foram aplicados problemas de oportunidade para máximo reuso e múltiplas fontes de água, envolvendo um contaminante e múltiplos contaminantes. Em todos os casos, observa-se a *corretude* (*correctness*) do MINEA 2.0 e a fidelidade do *software* ao algoritmo Diagrama de Fontes de Água. Será mostrado na execução dos exemplos que é possível sintetizar a Rede de Transferência de Massa para o caso de opção de regeneração com reciclo, mesmo não havendo essa função no *software*.

5.1. Wang e Smith (1994) – Máximo reuso

Problema exemplo extraído de Wang e Smith (1994) envolvendo um contaminante. Será aplicado para a oportunidade de máximo reuso. Será considerado que existe somente uma fonte de água externa disponível à concentração de 0ppm. Os dados limites do problema são apresentados na Tabela 5.1.

Tabela 5.1 - Dados do problema exemplo 5.1

Operação (k)	f_k(t/h)	Contaminante (i)	$C_{ik,max}$ (ppm)	$C_{fk,max}$ (ppm)
1	20	0	100	2
2	100	50	100	5
3	40	50	800	30
4	10	400	800	4

O resultado obtido pelo MINEA 2.0 é mostrado na Figura 5.1.

A partir do diagrama, observa-se, que o consumo de água primária foi de 90t/h e que o *pinch* de massa ocorre na concentração de 100ppm. O mesmo resultado foi obtido por Gomes (2002).

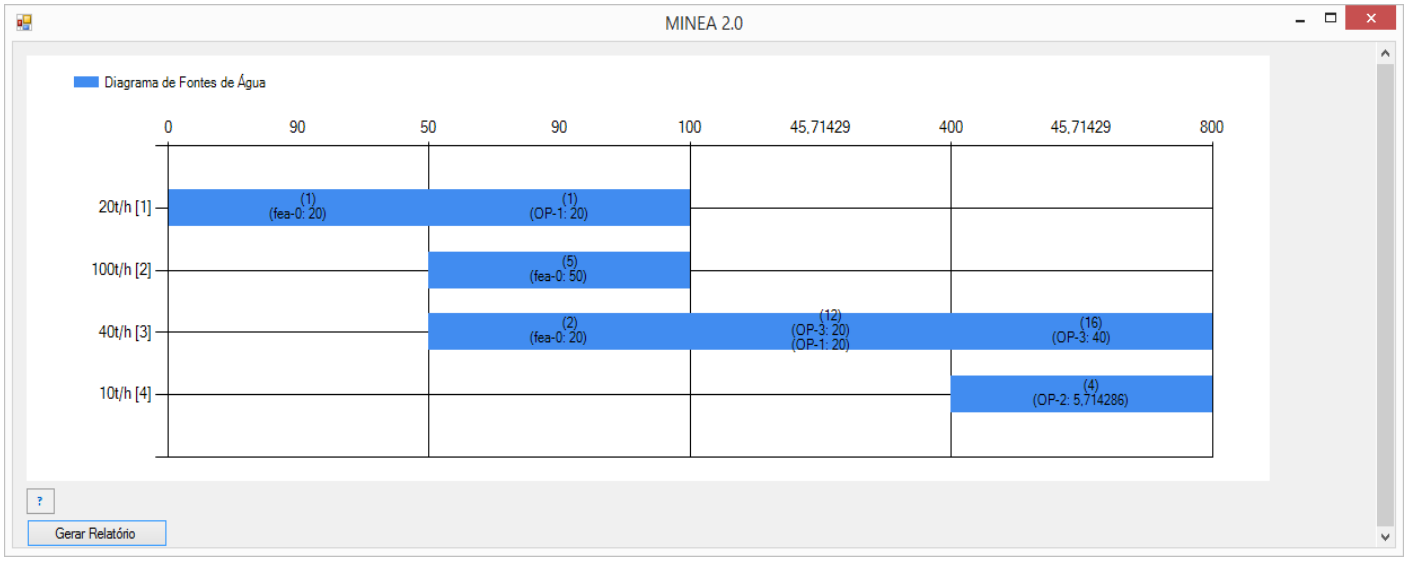

Figura 5.1 – Diagrama de Fontes de Água referente ao problema 5.2.1

A partir dos resultados obtidos é possível montar a Rede de Transferência de Massa para mínimo consumo de água primária (Figura 5.2).

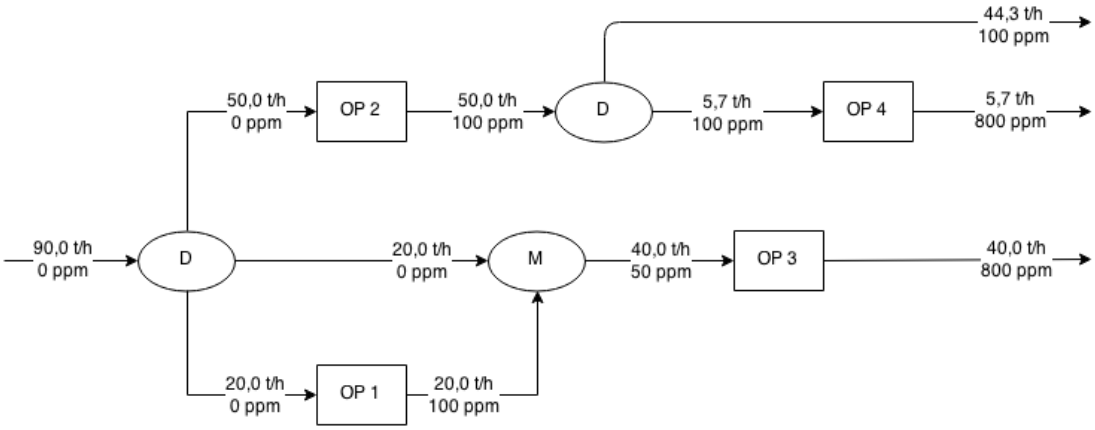

Figura 5.2 – Rede de Transferência de Massa correspondente ao diagrama da Figura 5.1.

5.2. Wang e Smith (1994) - Múltiplas Fontes de Água

Utilizando os mesmos dados da Tabela 4.1, mas agora considerando que há duas fontes de água externa: uma com concentração de 0ppm e a segunda com concentração de 25ppm. O resultado obtido é mostrado na Figura 5.3.

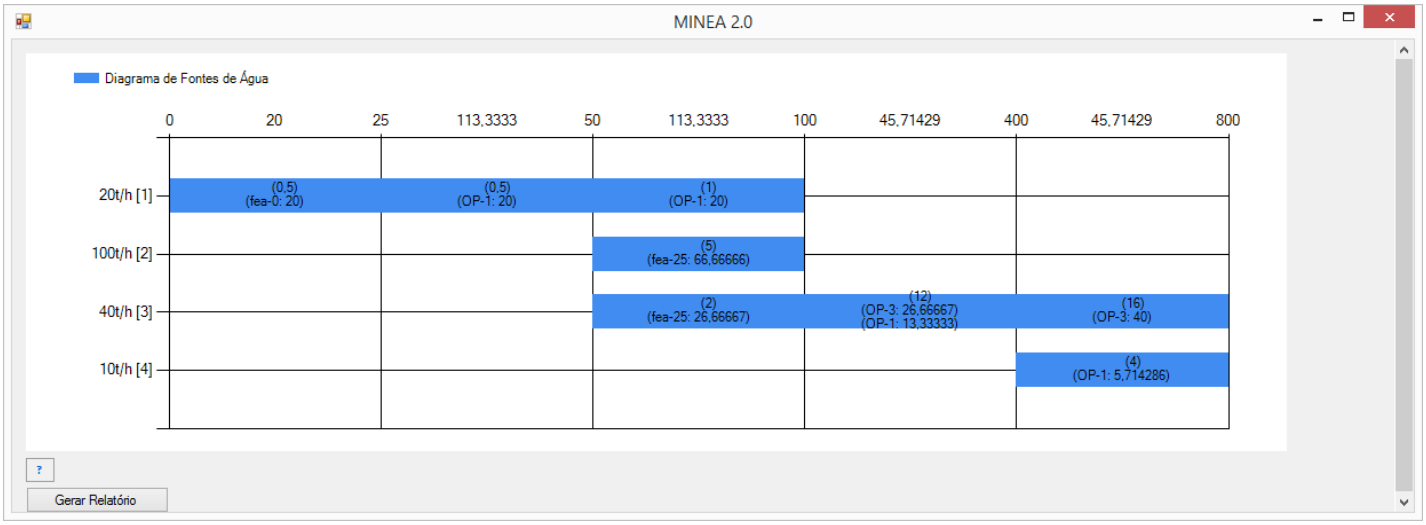

Figura 5.3 – Diagrama de Fonte de Água do exemplo 5.2.

A presença de uma fonte externa de menor qualidade (25ppm) levou a um consumo de água primária a 0ppm para 20t/h, uma redução significativa no consumo de água primária quando comparado na opção de máximo reuso. O consumo da fonte de 25ppm também é obtido do diagrama e é igual a 93,33t/h.

5.3. Wang e Smith (1994) - Regeneração com reciclo.

Na regeneração com reciclo, o efluente sofre tratamento para remoção dos contaminantes e pode ser utilizado no mesmo processo que o gerou. No algoritmo Diagrama de Fontes de Água, o processo de regeneração é considerado uma fonte interna de água com uma concentração de saída igual a C_{reg} e o procedimento algorítmico é similar ao utilizado para múltiplas fontes (Gomes, 2002).

Apesar do MINEA não apresentar uma opção de seleção de regeneração, os problemas de regeneração poderão ser efetuados utilizando a caixa de entrada de dados das fontes externas, conforme mostrado na Figura 5.4. Considera-se que a disponibilidade de um processo de regeneração com uma concentração de saída fixa C_{reg} e igual a 5ppm e uma fonte de externa de água a 0ppm. Ressalta-se que essa opção não é válida para problemas de regeneração com reuso, uma vez que problemas com essa opção apresentam vazão limitada. O

código-fonte não possui a função para esse tipo de problema mesmo que o usuário insira o limite de vazão em “Vazão Disponível”.

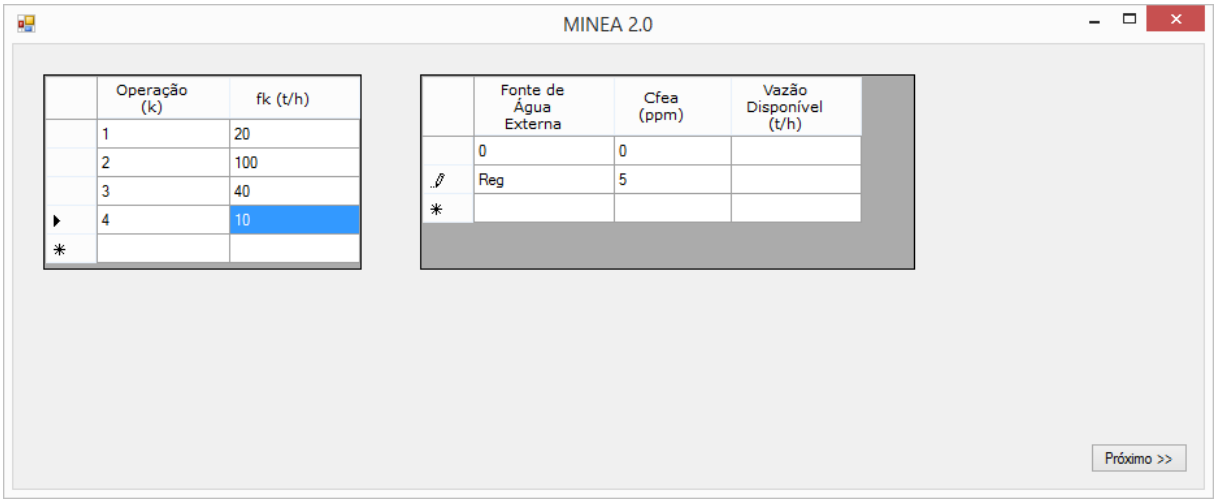

Figura 5.4 – Primeira tela de inserção de dados do MINEA 2.0 considerando regeneração com reciclo

Após a execução, o diagrama obtido é mostrado na Figura 5.5.

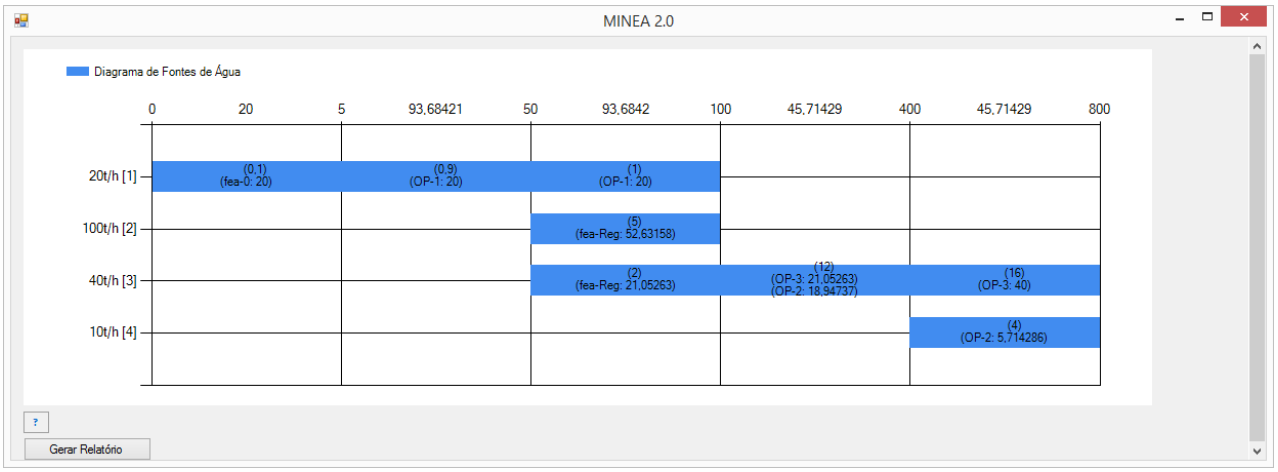

Figura 5.5 – Diagrama de Fontes de Água para o problema 5.3

Como pode ser observado a partir da Figura 5.5, por mais que o processo regenerativo tenha sido classificado como Fonte Externa, as Regra 1 de priorizar

as fontes internas continua sendo obedecida. A opção de regeneração com reciclo gera uma rede com consumo mínimo de água primária igual a 20t/h.

5.4. Wang e Smith (1994). Máximo Reuso

Problema exemplo para um sistema multicontaminante. Os dados foram extraídos de Wang e Smith (1994). De acordo com as regras apresentadas por Gomes *et al* (2013), a operação 1 é a operação de referência e o contaminante A é o contaminante de referência.

Considera-se que há somente uma fonte externa de água primária a 0ppm (0ppm de A, 0ppm de B, 0ppm de C). Os dados limites do problema são apresentados na Tabela. 5.2.

Tabela 5.2 – Dados do problema exemplo 5.4

Operação (k)	f_k**(t/h)**	**Contaminante (i)**	$C_{ik,max}$ **(ppm)**	$C_{fk,max}$ **(ppm)**	Δm_k**(kg/h)**
1	45	A	0	15	0,675
		B	0	400	18,000
		C	0	35	1,575
2	34	A	20	120	3,400
		B	300	12500	414,800
		C	45	180	4,590
3	56	A	120	220	5,600
		B	20	45	1,400
		C	200	9500	520,800

Executando o MINEA 2.0, são encontrados os resultados exibidos na Figura 5.6 e na Figura 4.14.

Observa-se a partir do Diagrama de Fontes de Água que o consumo mínimo de água primária alcançado pelo algoritmo foi 106,7t/h. Outros autores (Wang e Smith, 1994; Doyle e Smith, 1997) adotaram o mesmo problema exemplo em

seus trabalhos e obtiveram resultados bem próximos a esse valor. Os trabalhos que utilizaram programação matemática obtiveram valores de mínimo consumo ligeiramente menores que os trabalhos que utilizam uma abordagem conceitual.

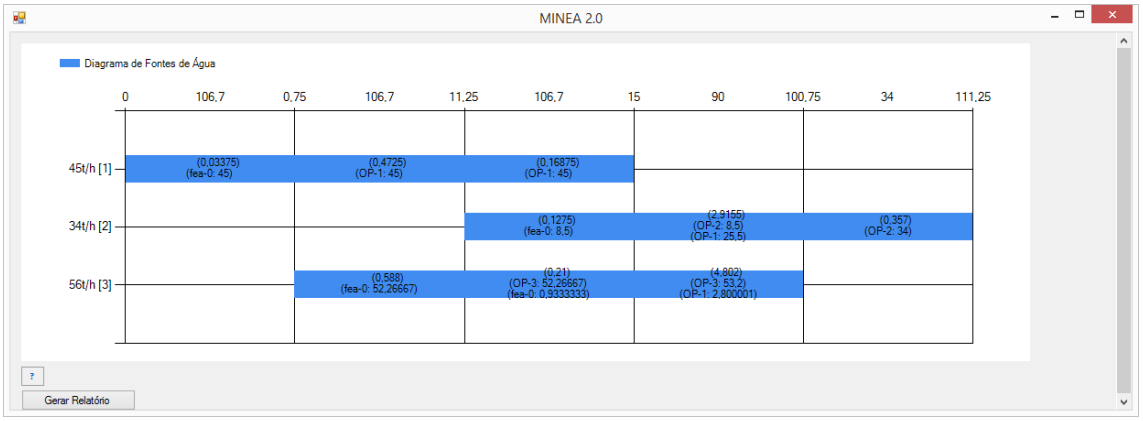

Figura 5.6 – Diagrama de Fontes de Água do exemplo 5.4

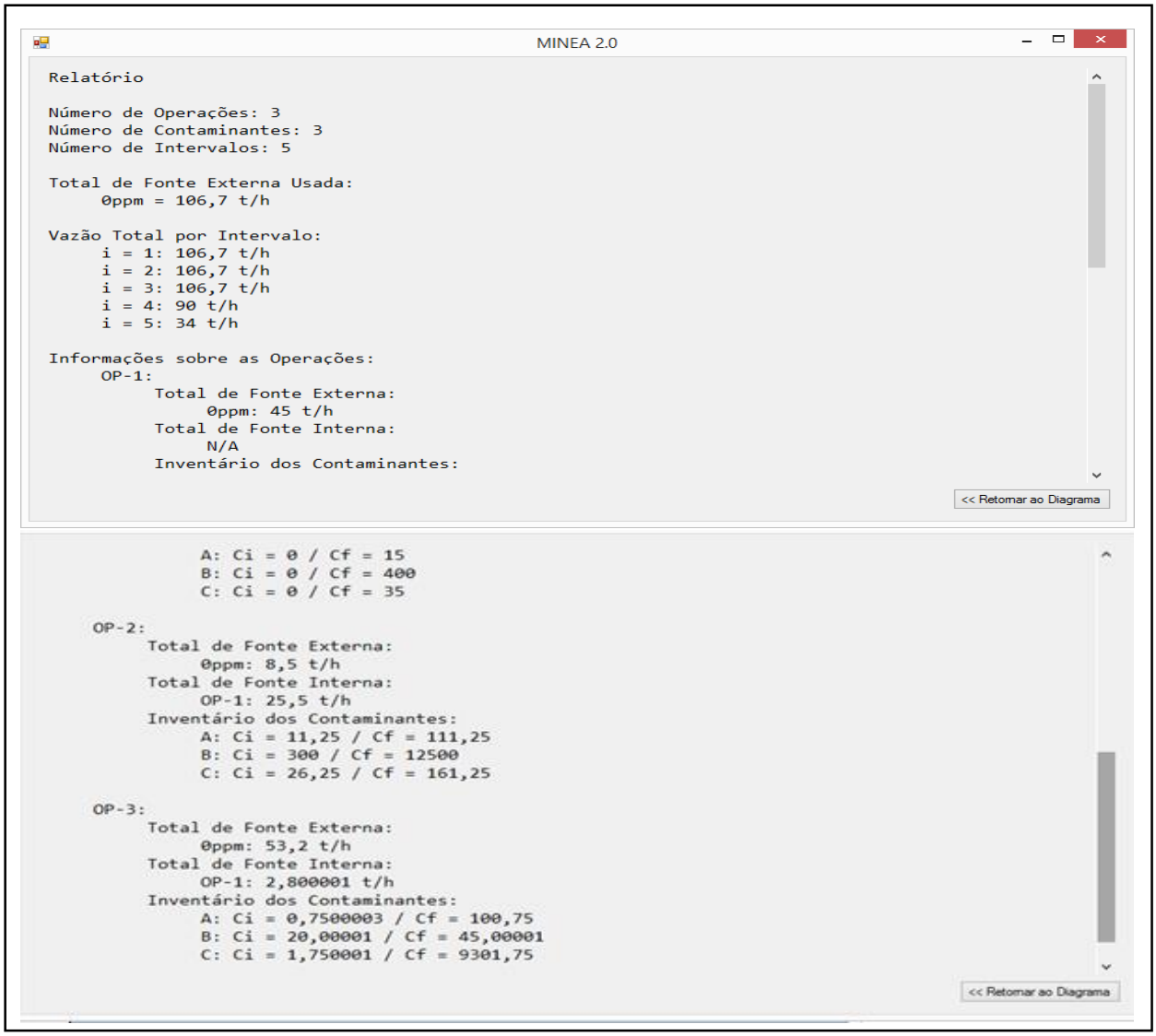

Figura 5.7 – “Relatório” correspondente ao diagrama da Figura 5.6

Para construir a Rede de Transferência de Massa referente ao Diagrama de Fontes de Água encontrado, os dados sobre as concentrações de cada contaminante podem ser retirados do “Relatório”, gerado pelo MINEA 2.0, que efetua as equações de balanço de massa e encontra os valores das concentrações de entrada e de saída de cada contaminante em cada operação. Observa-se que esse exemplo não apresenta violação de restrição e todos os valores das concentrações encontram-se dentro dos seus limites máximos e, portanto, não há necessidade de evoluir a rede.

5.5. Doyle e Smith, 1997 – Máximo Reuso.

Para mais um exemplo de sistemas multicontaminantes, a Tabela 5.3 apresenta os dados limites operacionais extraídos de Doyle e Smith (1997).

De acordo com as regras apresentadas por Gomes *et al* (2013), a operação 1 é a operação de referência e o contaminante A é o contaminante de referência.

Considera-se que há somente uma fonte externa de água primária a 0ppm (0ppm de A, 0ppm de B, 0ppm de C)

Tabela 5.3: Dados do problema exemplo 4.2.5

Operação (k)	f_k**(t/h)**	**Contaminante (i)**	$C_{ik,max}$ **(ppm)**	$C_{fk,max}$ **(ppm)**	Δm_k**(kg/h)**
1	34	A	0	160	5,44
		B	0	450	15,30
		C	0	30	1,02
2	75	A	200	300	7,50
		B	100	270	12,75
		C	500	740	18,00
3	80	A	300	800	40,00
		B	460	930	37,60
		C	400	900	40,00
4	20	A	600	1240	12,80
		B	850	1400	11,00
		C	390	1580	23,80

Executando o *software*, os valores encontrados são apresentados nas Figuras 5.8 3 5.9.

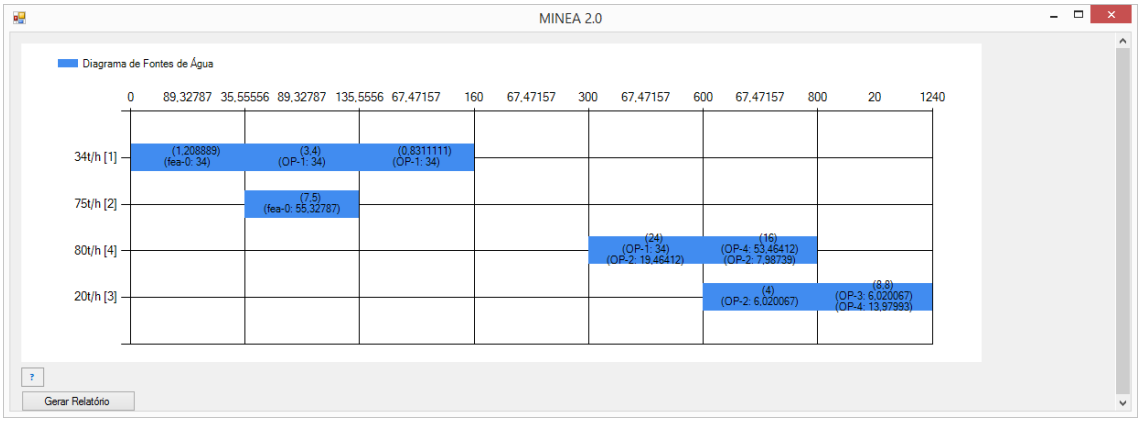

Figura 5.8 – Diagrama de Fontes de Água para o exemplo 4.5

O consumo de água primária foi de 89,33t/h, valor maior que o encontrado pelos autores (81,22t/h). Esse exemplo apresenta violação na operação 3 com o contaminante C. A concentração inicial encontrada pelo *software* foi igual a 508,2825ppm e a concentração final igual a 1.698,282ppm, ambas acima do limite máximo permitido e, portanto, essa rede deve ser evoluída para eliminar as violações.

MINEA 2.0

```
Relatório

Número de Operações: 4
Número de Contaminantes: 3
Número de Intervalos: 7

Total de Fonte Externa Usada:
     0ppm = 89,32787 t/h

Vazão Total por Intervalo:
     i = 1: 89,32787 t/h
     i = 2: 89,32787 t/h
     i = 3: 67,47157 t/h
     i = 4: 67,47157 t/h
     i = 5: 67,47157 t/h
     i = 6: 67,47157 t/h
     i = 7: 20 t/h

Informações sobre as Operações:
     OP-1:
          Total de Fonte Externa:
               0ppm: 34 t/h
          Total de Fonte Interna:
```

<< Retomar ao Diagrama

```
          Inventário dos Contaminantes:
               A: Ci = 0 / Cf = 160
               B: Ci = 0 / Cf = 450
               C: Ci = 0 / Cf = 30

     OP-2:
          Total de Fonte Externa:
               0ppm: 55,32787 t/h
          Total de Fonte Interna:
               N/A
          Inventário dos Contaminantes:
               A: Ci = 0 / Cf = 100
               B: Ci = 0 / Cf = 170
               C: Ci = 0 / Cf = 240

     OP-4:
          Total de Fonte Externa:
          Total de Fonte Interna:
               OP-1: 34 t/h
               OP-2: 19,46412 t/h
               OP-2: 7,98739 t/h
          Inventário dos Contaminantes:
               A: Ci = 133,1969 / Cf = 633,1969
```

<< Retomar ao Diagrama

```
     OP-3:
          Total de Fonte Externa:
          Total de Fonte Interna:
               OP-2: 6,020067 t/h
               OP-4: 13,97993 t/h
          Inventário dos Contaminantes:
               A: Ci = 472,7029 / Cf = 1112,703
               B: Ci = 606,8162 / Cf = 1156,816
               C: Ci = 508,2825 / Cf = 1698,282
```

<< Retomar ao Diagrama

Figura 5.9 – “Relatório” correspondente ao diagrama da Figura 5.8

5.6. Relvas *et al.* (2005) – Máximo Reuso

Para exemplo de máximo reuso, os dados limites estão apresentados na Tabela 5.4. Considera-se que há somente uma fonte de água externa a 0ppm. O resultado é apresentado na Figura 5.10.

Tabela 5.4 **– Dados do problema 5.6**

Operação (k)	f_k(t/h)	$C_{ik,max}$ (ppm)	$C_{fk,max}$ (ppm)
1	48	15	200
2	35	50	300
3	12	125	350
4	70	250	450

A partir da Figura 5.10, verifica-se que o consumo mínimo de água primária é de 78,08 t/h. Esse resultado foi o mesmo encontrado pelos autores, que utilizam a mesma metodologia abordada por Gomes (2002).

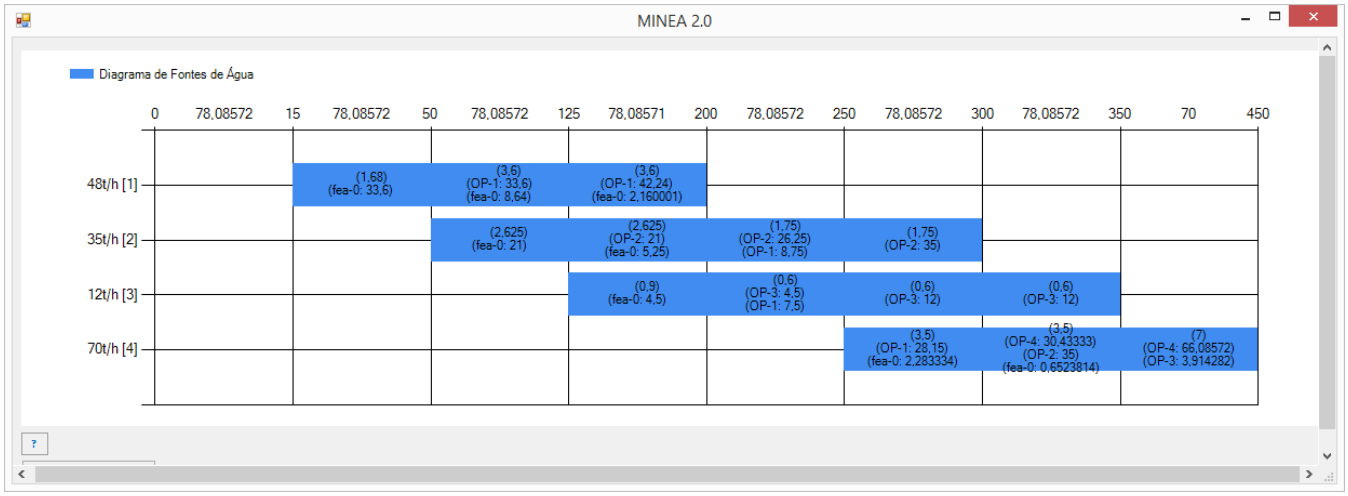

Figura 5.10 - Diagrama de Fontes de Água referente ao exemplo 5.6

5.7. Relvas *et al.* (2005) – Regeneração com Reciclo

Para esse exemplo, os dados limites estão apresentados na Tabela 5.4. Considera-se a disponibilidade de um processo de regeneração com uma

concentração de saída C_{reg} igual a 40ppm e que há somente uma fonte de água externa a 0ppm. O resultado é mostrado na Figura 5.11.

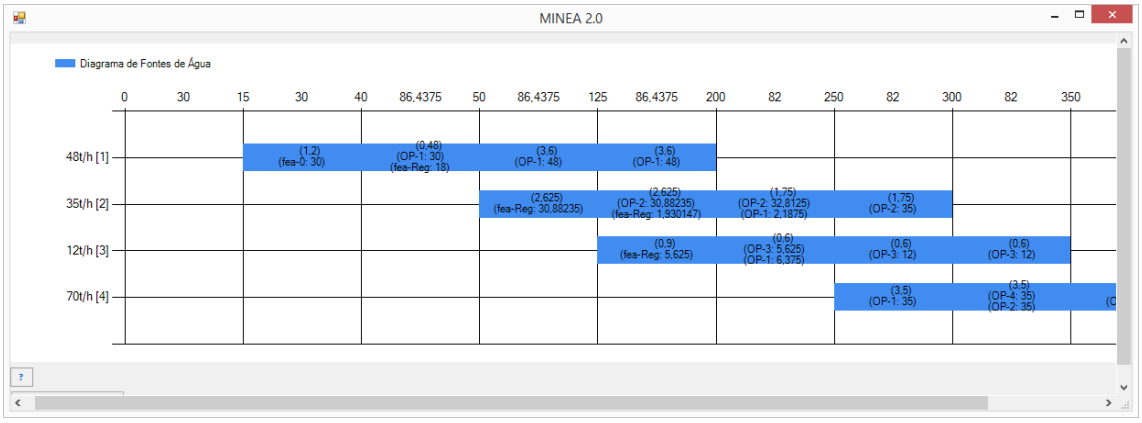

Figura 5.11 **-** Diagrama de Fontes de Água do exemplo 5.7**.**

Observa-se a partir da Figura 5.11 que o mínimo consumo de água primária foi igual a 30t/h, quando se considera a opção do processo regenerativo. O mesmo resultado foi encontrado por Relvas *et al*. (2005). Conclui-se que, a partir dos exemplos 5.3 e 5.7, a opção de regeneração com reciclo reduz significativamente o uso de água primária.

5.8. Prakotpol e Srinophakun (2004) – Máximo reuso

Para esse exemplo, os dados limites estão apresentados na Tabela 5.5.

Considera-se que há uma fonte de água externa a 0ppm. O resultado calculado pelo MINEA 2.0 é exibido na Figura 5.12.

Tabela 5.5 **–** Dados do problema 4.2.8

Operação (k)	f_k**(t/h)**	$C_{ik,max}$ **(ppm)**	$C_{fk,max}$ **(ppm)**
1	25	0	250
2	60	50	300
3	30	150	400
4	70	300	500

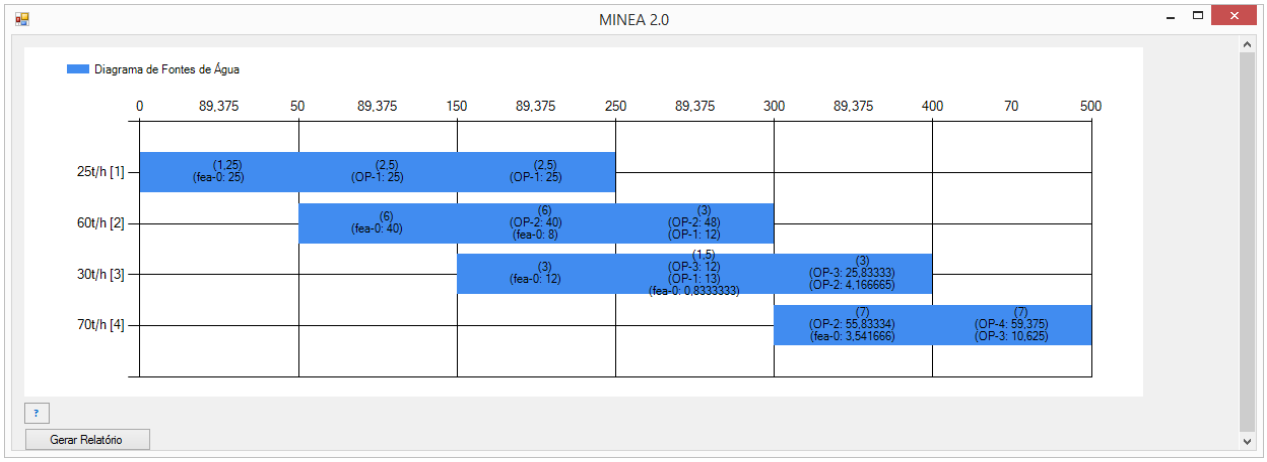

Figura 5.12 – Diagrama de Fontes de Água referente ao exemplo 5.8

O mínimo consumo de água encontrado pelo MINEA 2.0 para o problema 5.7 foi igual a 89,375t/h. O mesmo resultado foi encontrado pelos autores que utilizaram algoritmos genéticos para determinar a meta de mínimo consumo de água primária.

Os resultados acima exibidos, comprovam a habilidade do procedimento algorítmico Diagrama de Fontes de Água em resolver problemas de distribuição de água. Apesar de não garantir o ótimo global, o procedimento possibilita que as metas de mínimo consumo sejam alcançadas utilizando cálculos simples e, simultaneamente, gera a rede de transferência de massa correspondente. Os resultados podem ser aprimorados utilizando o resultado encontrado como estimativa inicial nos procedimentos que utilizam programação matemática e resolvidos em *solvers* disponíveis comercialmente

A partir dos exemplos apresentados nesta seção, prova-se a *corretude* funcional do *software* desenvolvido nesta tese. Todos os valores de mínimo consumo de água encontrados pelo MINEA 2.0 são os mesmos encontrados pela aplicação manual do algoritmo Diagrama de Fontes de Água. Isso demonstra que o código-fonte base poderá ser usado para evolução do *software*.

À exceção de exemplos extremamente complexos, o tempo de execução se mostrou menor que o tempo gasto inserindo os dados. Evidentemente, realizando os cálculos manualmente, o tempo se torna um fator importante quando os problemas possuem mais de um contaminante, pois a etapa de

deslocamento das concentrações exige a análise de cada uma das concentrações.

O *software* foi testado e apresentou estabilidade nas versões do Windows® 7, 8 e 8.1.

Capítulo 5 - Conclusão

O *software* desenvolvido neste trabalho apresenta boa *corretude* funcional ao ser executado. Ele foi construído de forma a ser um *software* embrião para outras funções serem implementadas e, por isso, a base de todo o código é orientado a executar precisamente o procedimento algorítmico Diagrama de Fontes de Água para opção de máximo reuso.

O código-fonte foi programado seguindo a programação orientada a objeto, o que possibilita melhor reaproveitamento do código e gerenciamento das classes e funções.

Para suprir a falta da visualização da Rede de Transferência de Massa, foi criado um Relatório que contém todas as informações, em forma de texto, necessárias para a construção da rede.

O procedimento algorítmico Diagrama de Fontes de Água abordado nesse trabalho gera resultados satisfatórios e que, apesar de não garantir o ótimo global, é uma ferramenta da fácil implementação e que gera simultaneamente a Rede de Transferência de Massa. O resultado pode ser refinado e os valores encontrados por essa metodologia serem utilizados como estimativa inicial em *solvers* ou simuladores.

As próximas etapas a serem desenvolvidas no MINEA 2.0 incluem:

- Desenvolvimento de uma comunicação com solvers comerciais;
- Apresentar visualmente a Rede de Transferência de Massa no modo clássico;
- Encontrar as violações da rede e automaticamente corrigi-las.

AMPL. Disponível em: <ampl.com>. Acesso em: setembro/2014

ANTONIOU, A; LU, WS. **Practical Optimization: Algorithms and Engineering Applications**. Springer US. 2007. 670p.

Aspen Tech. Disponível em: <http://www.aspentech.com/> Acessado em: setembro/2014

BAGAJEWICZ, M., "Review of recent design procedures for water networks in refineries and process plants". **Computers and Chemical Engineering**, 24, 2093-2113, 2000.

BELFIORE, P., FÁVERO, L.P. **Pesquisa Operacional para cursos de engenharia**. Rio de Janeiro. Elsevier. 2013

CASTRO, P., MATOS, H., FERNANDES, M.C., PEDRO NUNES, C. "Improvements for mass-exchange networks design". **Chemical Engineering Science**, 54, 1649-1665, 1999.

CHEMECAD. Disponível em: <www.chemstations.com/ > Acesso em: setembro/2014

CPLEX. Disponível em: <http://www-01.ibm.com/> Acesso em: setembro/2014

DOYLE, S.J., SMITH, R.."Targeting water reuse with multiple contaminants" **Transactions of International of Chemical Engineering**, 75 (B), 181-189, 1997. DUNN e EL-HALWAGI, 2003

DUNN,R.F., EL-HALWAGI, M.M., "Process integration technology review: background and applications in the chemical process industry", **Journal of Chemical Technology and Biotechnology**, 78 (9), 1011-1021, 2003.

EL-HALWAGI, M.M., MANOUSIOUTHAKIS, V. "Synthesis of mass exchange networks", **American Institute of Chemical Engineering Journal**, 35 (8), 1233-1244, 1989. EL-HALWAGI, 1998

EL-HALWAGI, M.M. "Pollution prevention through process integration". **Clean Products and Process** *1*, 5-19, 1998

FARIA D. C., BAGAJEWICZ, M. "Comparative Analysis of different assumptions for the Design of Single-Contaminant Water Networks". **Chemical Engineering Communications**, Vol. 197, No 6, pp. 859–880, 2010

FARIA D. C., BAGAJEWICZ, M. "On the Appropriate Modeling of Process Plant Water Systems". **AIChE Journa**l, 56, 3, pp. 668-689, 2010

FOO, D.C.Y. “State-of-the-Art Review of *Pinch* Analysis Techniques for Water Network Synthesis”. **Ind. Eng. Chem***. Res.***, *48,* 5125–5159, 2009

GOMES, J.F.S., **Procedimento para minimização de efluentes aquosos**, Dissertação de M.Sc., TPQBq, EQ/UFRJ, Rio de Janeiro, 2002.
GOMES, J.F.S., MIRRE, R.C., DELGADO, B.E.P.C, QUEIROZ, E.M., PESSOA, F.L.P. “Water Sources Diagram in Multiple Contaminant Processes: Maximum Reuse”. **Industrial & Engineering Chemestry Research,** 52, 4, 1667-1677, 2013

Gundersen (2002). “Process Integration Primer”. International Energy Agency. Disponível em: <www.iea-pi.org>

Matlab. Disponível em: <http://www.mathworks.com/>

GAMS. Disponível em: <www.**gams**.com/ >

HoneyWell. Disponível em: www.honeywellprocess.com/

HYSYS. Disponível em: <www.**aspen**tech.com/**hysys**/>

Lindo System. Disponível em: <www.**lindo**.com>

JEZOWSKI, J. “Review of Water Network Design Methods with Literature Annotations”. **Ind. Eng. Chem***. Res*., 49, 4475–4516, 2010

Jornal El Pais – Brasil. Disponível em: <brasil.elpais.com>. Acesso em: janeiro/2015

Lam, H.L, Klemes, J.J, Kravanja, z., Varbanov, P.S. *“*Software tools overview: process integration, modelling and optimisation for energy saving and pollution reduction”. **Asia-Pac. J. Chem. Eng**. 6: 696–712, 2011

MANN, J.G.Jr. **Process Integration: Unifying Concepts, Industrial Applications and *Software* Implementation**. Tese D.Sc. Faculty of the Virginia Polytechnic Institute and State University. 1999
Blacksburg, VA, 1999

MIERZWA, J.C. **O uso racional e o reuso como ferramentas para o gerenciamento de água e e efluentes na indústria: estudo de caso da Kodak brasileira (Volumes 1 e 2)**. Tese de D.Sc. Escola Politécnica da Universidade de São Paulo. São Paulo, 2002

NG, D.K.S, CHEW, I.M.L, TAN, R.R.T, FOO, D.C.Y, OOI, M.B.L, EL-HALWAGI, M.M. “RCNet: An optimisation *software* for the synthesis of resource conservation networks”. **Process Safety and Enviromental Protection***,* 92, 917 – 928, 2014

ONU Water. Disponível em: www.un**water**.org/. Acesso em: março/2014

OSL. Disponível em: < http://www-01.ibm.com/>. Acesso em: fevereiro/2015

Operador Nacional do Sistema Elétrico – ONS. Disponível em: www.**ons**.org.br/. Acesso em: fevereiro/2015

PERLINGEIRO, C.A.G., **Engenharia de Processos: análise, simulação, otimização e síntese de processos químicos**, São Paulo: Ed. Edgard Blücher, 2005.

Perry´s Chemical Engineers Handbook. 8ª ed. McGraw-Hill.2008

Pollution Engineering – BR. Disponível em: <www.revistape.com.br>

Prakotpol, D., Srinophakun T. GA*Pinch*: genetic algorithm toolbox for water *pinch* Technology. **Chemical Engineering and Processing,** Vol. 43, p 203–217, 2004.

PRO/II. Disponível em: <iom.invensys.com>

PUIGJANER, L., ESPUÑA, A., ALMATÓ, M. "A software tool for helping in decision-making about water management in batch process industries". **Waste Management,** 20, 645-649, 2000.

QUEIROZ, E.M., PESSOA, F.L.P. "Integração de Processos: uma ferramenta para minimizar o consumo energético e o impacto ambiental". **Fronteiras da Engenharia Química 1**, Série Escola Piloto de Engenharia Química, PEQ/COPPE/UFRJ, Rio de Janeiro: Ed. e-papers, 2005.

Relvas, S., Matos, H. A., Fernandes, M. C., CASTRO, P. Nune**s**, C. P. "AquoMin: An Approach to Pollution Prevention based on *Pinch* Analysis". **Computer Aided Chemical Engineering**, 18, 475 – 480, 2004.

Relvas, S., Matos, H. A., Fernandes, M. C., Nunes, C. P. "AquoMin: Targeting and Design of Mass-Exchange Networks featuring Regeneration Recycle". **Computer Aided Chemical Engineering***,* 20, 685–690, 2005

Relvas, S., Matos, H. A., Fernandes, M. C., CASTRO, P. Nunes, C. P. "AquoMin: A *software* tool for Mass-Exchange Networks targeting and design". **Computers & Chemical Engineering**, 6, 1085 – 1105, 2008

Revista Valor Econômico. Disponível em: <www.valor.com.br>. Acesso em: 30/01/2015

Sabesp. Disponível em: < www.**sabesp**.com.br/>. Acesso em: em 30/01/15

SANTOS, R. P. **Extensão e Automatização do Diagrama de Fontes de Água Usado no Gerenciamento e Redução do Consumo de Água em Plantas Industriais**. Dissertação (Mestrado em Tecnologia de Processos Químicos e Bioquímicos), Escola de Química -Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2007

SAVELSKI, M., BAGAJEWICZ, M. "On the necessary conditions of optimality of water utilization systems in process plants with multiple contaminants". **Chemical Engineering Science**. 58, 5349-5362, 2003.

SALVESKI, M.J., BAGAJEWICZ, M. "On the optimality conditions of water utilization systems in process plants with single contaminants". **Chemical Engineering Science**, 55, 5035-5048, 2000a.

SAVELSKI, M., BAGAJEWICZ, M. "Design of water utilization systems in process plants with a single contaminant". **Waste Management**, 20, 659-664, 2000b.

SHIKLOMANOV, I. A. "World Water Resources". Disponível em: < http://webworld.unesco.org/water/ihp/db/shiklomanov/>

SMITH, R. **Chemical Process Design and Integration**. John Wiley & Sons Ltd, 2005.

STEIMEL, J., HARRMANN, M., SCHEMBECKER, G., ENGELL, S. "A framework for the modeling and optimization of process superstructures underuncertainty" **Chemical Engineering Science**, 115, 225 – 237, 2014

TELES, J.P, CASTRO, P.M., NOVAIS, A.Q. "OptWatNet - A *Software* for the Optimal Design of Water-Using Networks with Multi-contaminants". **17th European Symposium on Computer Aided Process Engineering**. 2007

ULLMER, C., KUNDE, N., LASSAHN, A., GRUHN, G., SCHULZ, K. WADOTM: "Water design optimization – methodology and *software* for the synthesis of process water systems". **Journal of Cleaner Production**, 13, 485-494, 2005.

US Environmental Protection Agency. Disponível em: **<**http://www.epa.gov/p2/>. Acesso em: set/2014

WANG, Y.P., SMITH, R. "Wastewater minimisation". **Chemical Engineering Science**, 49 (7), 981-1006, 1994a.

Water Design. Disponível em: <http://www.design.che.vt.edu/waterdesign/waterdesign.html. Acesso em: setembro/2014

WATER. Disponível em: <http://www.ceas.manchester.ac.uk/>

Anexo A.1

O diagrama representado na Figura A.1.1, demonstra a estrutura final do código-fonte, bem como o relacionamento entre as classes que o compõe. Nele podemos verificar que as classes podem ser agrupadas em três conjuntos: (i) formulários de visualização; (ii) entidades; e (iii) classes de cálculos.

Os formulários representam as interfaces com as quais o usuário interage; as entidades são classes responsáveis por armazenar e transportar as informações; e as classes de cálculos são as responsáveis por manipular as informações presentes dentro das entidades.

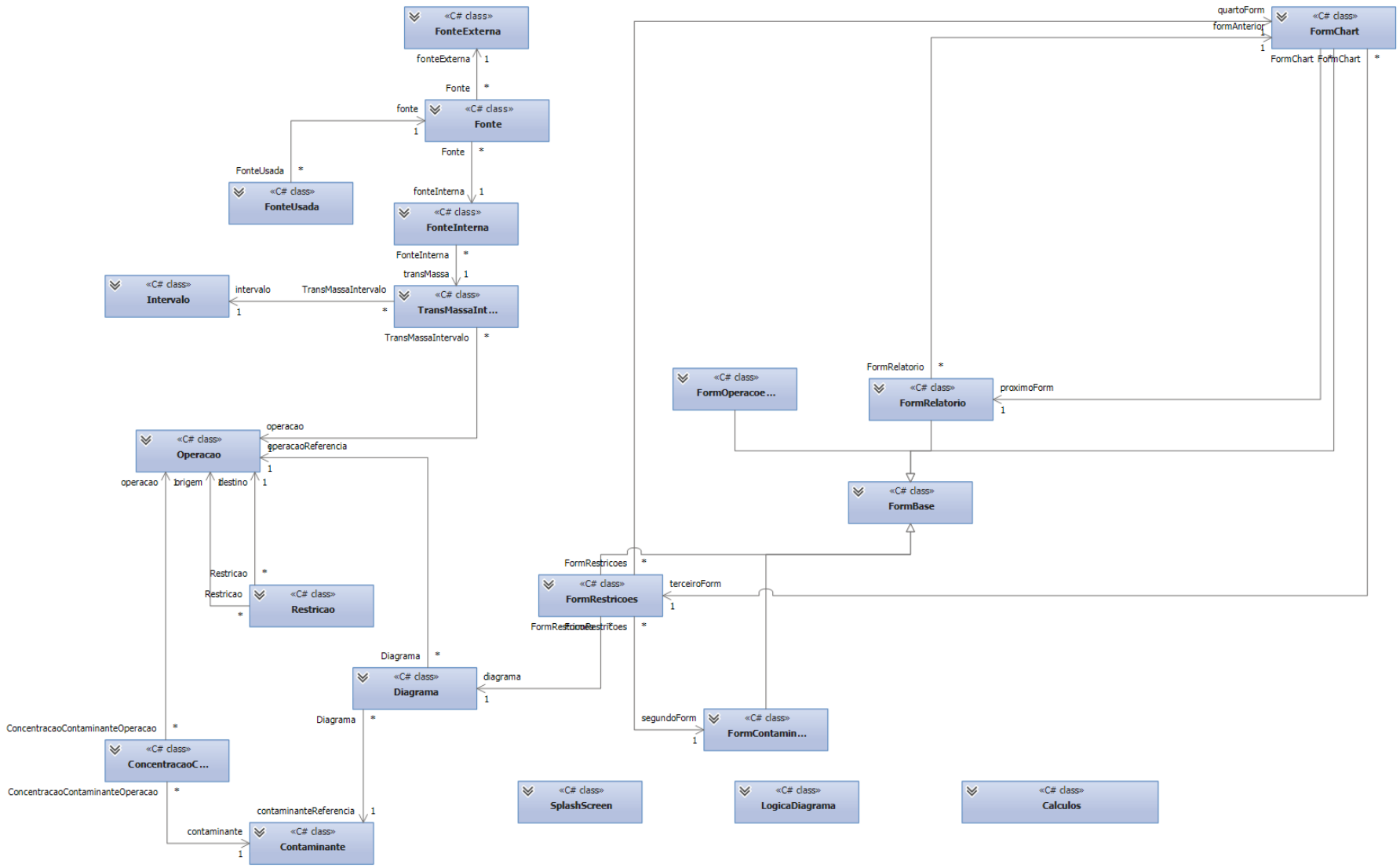

Figura A.1.1 – Modelo Lógico do MINEA 2.0.

Anexo A.2

Arquivo Calculos.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using DFA_Teste0.entidades;
using System.Windows.Forms;


namespace DFA_Teste0 {
    class Calculos {
        public static float calculoDeltaM (float fk, float ce, float cf) {

            float resultado = 0;

            resultado = (fk * (cf - ce));

            return resultado;
        }


        //Dados de entrada: nº de operações, Ce, Cf, deltaM, fonte

        public static void calculoDiagramaFontesAgua(FormRestricoes
formRestricoes) {
            Diagrama diagrama = new Diagrama();
            formRestricoes.diagrama = diagrama;
            DataGridView gridOperacao =
(DataGridView)formRestricoes.segundoForm.primeiroForm.Controls.Find("dataGridView
1", false)[0];
            DataGridView gridContaminantes =
(DataGridView)formRestricoes.segundoForm.Controls.Find("dataGridView1",
false)[0];
            DataGridView gridFontesExternas =
(DataGridView)formRestricoes.segundoForm.primeiroForm.Controls.Find("dataGridView
2", false)[0];
            DataGridView gridRestricoes =
(DataGridView)formRestricoes.Controls.Find("dataGridViewMatriz", false)[0];

            diagrama.qtdContaminantes =
formRestricoes.segundoForm.qtdContaminantes;

            // Recuperar  todas as infos necessárias para realizar os cálculos e
preencher o diagrama.
            //Os métodos preenche estão no final.
            // 1/4 - Recuperar o número de operações k, Cek, Cfk, deltaMk, fk
            // 2/4 - Recuperar concentrações dos contaminantes do segundoForm
            // 3/4 - Recuperar o número de fontes externas disponíveis e suas
concentrações
            // 4/4 - Recuperar restrições de reuso entre as operações
            preencheOperacoes(diagrama, gridOperacao,
formRestricoes.segundoForm.qtdContaminantes,
formRestricoes.segundoForm.comboBox1.Text);
```

```
            preencheConcentracoesContaminantes(diagrama, gridContaminantes,
formRestricoes.segundoForm.qtdContaminantes,
formRestricoes.segundoForm.comboBox2.Text);
            preencheFontesExternas(diagrama, gridFontesExternas);
            preencheRestricoes(diagrama, gridRestricoes);

            ajusteMultiplosContaminantes(diagrama);

            //Array com com todas as concentrações recuperadas
            //Excluir as concentrações duplicatas
            //Arrumar em ordem crescente de concentração
            diagrama.concentracoes.Sort();

            //Definir os intervalos de concentração i = nº de concentrações
arrumadas - 1
            defineIntervalos(diagrama);

            //if (diagrama.contaminantes.Count == 1) {
                //Criar os transMassas de cada intervalo de cada operacao
                defineTransMassaIntervalosParaUmContaminante(diagrama);
                defineAlocacaoParaUmContaminante(diagrama);



        }

        private static void ajusteMultiplosContaminantes(Diagrama diagrama) {
            if (diagrama.qtdContaminantes > 1) {

                //fixarFontesExternas(diagrama);
                ajusteConcentracoesOperacaoReferencia(diagrama);
                ajusteConcentracoesContaminanteReferencia(diagrama);
                ajusteListaConcentracoes(diagrama);
            }
        }

        private static void ajusteListaConcentracoes(Diagrama diagrama) {
            diagrama.concentracoes.Clear();

            foreach (Operacao operacao in diagrama.operacoes.Values) {
                foreach (ConcentracaoContaminanteOperacao cco in
operacao.listaConcentracoes) {
                    if
(cco.contaminante.nome.Equals(diagrama.contaminanteReferencia.nome)){
                        if
(!diagrama.concentracoes.Contains(cco.concentracaoEntrada)) {
                            diagrama.concentracoes.Add(cco.concentracaoEntrada);
                        }

                        if
(!diagrama.concentracoes.Contains(cco.concentracaoSaida)) {
                            diagrama.concentracoes.Add(cco.concentracaoSaida);
                        }
                    }
                }
            }

            for (int i = 0; i < diagrama.fontesExternas.Values.ToArray().Length;
i++) {
                FonteExterna fe = diagrama.fontesExternas.Values.ToArray()[i];
                if (!diagrama.concentracoes.Contains(fe.concentracao)) {
                    diagrama.concentracoes.Add(fe.concentracao);
```

```
                }
            }
        }

        private static void ajusteConcentracoesContaminanteReferencia(Diagrama
diagrama) {
            foreach (Operacao operacao in diagrama.operacoes.Values.ToArray()){
                foreach (ConcentracaoContaminanteOperacao cco in
operacao.listaConcentracoes) {
                    if (!cco.contaminante.referencia && !operacao.referencia) {
                        ConcentracaoContaminanteOperacao ccoref =
buscaConcentracao(diagrama.operacaoReferencia, cco.contaminante.nome);
                        if (cco.concentracaoEntrada < ccoref.concentracaoSaida) {
                            float concentracaoEntradaAjustada =
                                (cco.concentracaoEntrada
                                    - ccoref.concentracaoEntrada
                                    + buscaConcentracao(
                                        diagrama.operacaoReferencia,
                                        diagrama.contaminanteReferencia.nome)
                                    .concentracaoEntrada
                                    * diagrama.constanteTransferenciaLinear(
                                        cco.contaminante,
                                        diagrama.operacaoReferencia))
                                / diagrama.constanteTransferenciaLinear(
                                    cco.contaminante,
                                    diagrama.operacaoReferencia);

                            float concentracaoSaidaAjustada =
                                (cco.concentracaoSaida
                                - cco.concentracaoEntrada
                                + concentracaoEntradaAjustada*
                                diagrama.constanteTransferenciaLinear(
                                    cco.contaminante,
                                    cco.operacao))
                                /diagrama.constanteTransferenciaLinear(
                                    cco.contaminante,
                                    cco.operacao);

                            ConcentracaoContaminanteOperacao
ccoContaminanteReferenciaOperacaoAtual = buscaConcentracao(cco.operacao,
diagrama.contaminanteReferencia.nome);

ccoContaminanteReferenciaOperacaoAtual.concentracaoEntrada =
concentracaoEntradaAjustada;

ccoContaminanteReferenciaOperacaoAtual.concentracaoSaida =
concentracaoSaidaAjustada;
                        }
                    }
                }
            }
        }

        private static ConcentracaoContaminanteOperacao
buscaConcentracao(Operacao operacao, string nomeContaminante) {
            foreach (ConcentracaoContaminanteOperacao cco in
operacao.listaConcentracoes) {
                if (cco.contaminante.nome.Equals(nomeContaminante)) {
                    return cco;
                }

            }
```

```
            return null;
        }

        private static void ajusteConcentracoesOperacaoReferencia(Diagrama
diagrama) {
            float menorConcentracaoFonteExterna =
diagrama.menorConcentracaoFonteExterna();
            foreach (ConcentracaoContaminanteOperacao cco in
diagrama.operacaoReferencia.listaConcentracoes) {
                if (cco.concentracaoEntrada != menorConcentracaoFonteExterna) {
                    float deltaMOriginal = calculoDeltaM(cco.operacao.fk,
cco.concentracaoEntrada, cco.concentracaoSaida);
                    cco.concentracaoEntrada = menorConcentracaoFonteExterna;
                    cco.concentracaoSaida = (deltaMOriginal - (cco.operacao.fk *
menorConcentracaoFonteExterna)) / cco.operacao.fk;
                }
            }
        }


        //Função Alocação
        private static void defineAlocacaoParaUmContaminante(Diagrama diagrama) {
            for (int i = 0; i < diagrama.intervalos.Count; i++) {
                defineAlocacaoIntervalo(diagrama, i);
            }
        }


        //Função Alocação para cada intervalo. Repete a mesma lógica para todos
os intervalos
        private static void defineAlocacaoIntervalo(Diagrama diagrama, int
posicaoIntervalo) {
            Intervalo intervalo = diagrama.intervalos[posicaoIntervalo];
            Intervalo intervaloAnterior = null;

            //Jogar do relatórioFontes para o fontesDisponíveis
            if (posicaoIntervalo != 0) {
                intervaloAnterior = diagrama.intervalos[posicaoIntervalo - 1];
                String[] keys = intervaloAnterior.relatorioFontes.Keys.ToArray();
                for (int i = 0; i < keys.Length; i++) {
                    intervalo.fontesDisponiveis.Add(keys[i],
intervaloAnterior.relatorioFontes[keys[i]]);
                }
            }

           //Colocar as fontes externas em seus devidos lugares
            String[] chaves = diagrama.fontesExternas.Keys.ToArray();
            for (int i = 0; i < diagrama.fontesExternas.Count; i++) {
                if (diagrama.fontesExternas[chaves[i]].concentracao <=
intervalo.concentracaoInicial
                    && !intervalo.fontesDisponiveis.Keys.Contains(chaves[i])) {
                    Fonte fonte = new Fonte();
                    fonte.fonteExterna = diagrama.fontesExternas[chaves[i]];
                    if (diagrama.fontesExternas[chaves[i]].vazao < 0) {
                        fonte.vazaoDisponivel = 1000000000000000;
                    }
                    else fonte.vazaoDisponivel =
diagrama.fontesExternas[chaves[i]].vazao;


                    intervalo.fontesDisponiveis.Add(chaves[i], fonte);
                }
```

```
            }

            //Começando a lógica...

            //1º bloco: Se trasmassa|i está em i - 1: alocar todas
            if (posicaoIntervalo > 0) {
                chaves = intervalo.transmassas.Keys.ToArray();
                for (int i = 0; i < intervalo.transmassas.Count; i++) {
                    TransMassaIntervalo transmassa =
intervalo.transmassas[chaves[i]];
                    String key = transmassa.operacao.nomeOperacao + "-" +
intervaloAnterior.concentracaoFinal;
                    if(intervaloAnterior.transmassas.ContainsKey(key)) {
                        TransMassaIntervalo transmassaAnterior =
intervaloAnterior.transmassas[key];
                        alocarVazao(transmassa, transmassaAnterior.key());
                    }
                }
            }
            //2º bloco: uso das fontes internas
            chaves = intervalo.transmassas.Keys.ToArray();
            for (int i = 0; i < intervalo.transmassas.Count; i++) {
                TransMassaIntervalo transmassa =
intervalo.transmassas[chaves[i]];
                Fonte origem = null;
                while (transmassa.deltaM() - transmassa.deltaMtemp() > 0 &&
(origem = buscaFonteInternaDeMaiorConcentracao(intervalo.fontesDisponiveis,
diagrama.restricoes, transmassa)) !=null) {
                    alocarVazao(transmassa, origem.key());
                }

            }
            //3º bloco: uso fontes externas
            chaves = intervalo.transmassas.Keys.ToArray();
            for (int i = 0; i < intervalo.transmassas.Count; i++) {
                TransMassaIntervalo transmassa =
intervalo.transmassas[chaves[i]];
                if (transmassa.deltaM() - transmassa.deltaMtemp() > 0) {
                    Fonte origem =
buscaFonteExternaDeMaiorConcentracao(intervalo.fontesDisponiveis);

                    if (origem != null) {
                        alocarVazao(transmassa, origem.key());
                        if
(!intervalo.fontesExternasUsadas.Keys.Contains(origem.key())) {
                            FonteUsada fonteUsada = new FonteUsada();
                            fonteUsada.fonte =
transmassa.fontesUsadas[origem.key()].fonte;
                            intervalo.fontesExternasUsadas.Add(origem.key(),
fonteUsada);
                        }
                        intervalo.fontesExternasUsadas[origem.key()].vazaoUsada
+= transmassa.fontesUsadas[origem.key()].vazaoUsada;
                    }
                }

            }

            //Relatório: tabela do tipo Fonte
            chaves = intervalo.fontesDisponiveis.Keys.ToArray();
            for (int i = 0; i < chaves.Length ; i++) {
```

```
                intervalo.relatorioFontes.Add(chaves[i],
intervalo.fontesDisponiveis[chaves[i]]);
            }

            chaves = intervalo.transmassas.Keys.ToArray();
            for (int i = 0; i < chaves.Length ; i++) {
                TransMassaIntervalo temp = intervalo.transmassas[chaves[i]];
                Fonte fonteRelatorio = new Fonte();
                fonteRelatorio.fonteInterna = new FonteInterna();
                fonteRelatorio.fonteInterna.transMassa = temp;
                fonteRelatorio.fonteInterna.vazaoDisponivel =
temp.vazaoParaDeltaM();
                fonteRelatorio.fonteInterna.concentracao =
temp.intervalo.concentracaoFinal;
                fonteRelatorio.vazaoDisponivel = temp.vazaoParaDeltaM();
                intervalo.relatorioFontes.Add(fonteRelatorio.key(),
fonteRelatorio);
            }



        }


        private static Fonte
buscaFonteInternaDeMaiorConcentracao(Dictionary<String, Fonte>
tabelaFontesDisponiveis, Dictionary<String, Restricao> restricoes,
TransMassaIntervalo transmassa) {
            Fonte resultadoBusca = null;
            for (int i = 0; i < tabelaFontesDisponiveis.Count; i++) {

                Fonte fonte =
tabelaFontesDisponiveis[tabelaFontesDisponiveis.Keys.ToArray()[i]];
                if (fonte.fonteInterna != null) {
                    Restricao restricao =
restricoes[fonte.fonteInterna.transMassa.operacao.nomeOperacao + "-" +
transmassa.operacao.nomeOperacao];
                    if (fonte.vazaoDisponivel > 0
                        && (restricao.pode)
                        && (resultadoBusca == null
                            || (fonte.concentracao() >=
resultadoBusca.concentracao()
                                )) {
                        resultadoBusca = fonte;
                    }
                }
            }

            return resultadoBusca;
        }

        private static Fonte
buscaFonteExternaDeMaiorConcentracao(Dictionary<String, Fonte>
tabelaFontesDisponiveis) {
            Fonte resultadoBusca = null;
            for (int i = 0; i < tabelaFontesDisponiveis.Count; i++) {
                Fonte fonte =
tabelaFontesDisponiveis[tabelaFontesDisponiveis.Keys.ToArray()[i]];
                if (fonte.fonteInterna == null && (resultadoBusca == null ||
fonte.concentracao() > resultadoBusca.concentracao())) {
                   resultadoBusca = fonte;
                }
```

```
            }
            return resultadoBusca;
        }



        private static float calculoParaAbsorverDeltaM(float deltaM, float
deltaMJaAlocado, float cfTransmassa, float cFonteDisponivel, float
vazaoDisponivel) {
            float deltaMRequerido = deltaM - deltaMJaAlocado;
            float vazaoCalculada = deltaMRequerido / (cfTransmassa -
cFonteDisponivel);
            if (vazaoCalculada > vazaoDisponivel) {
                return vazaoDisponivel;
            }
            else {
                return vazaoCalculada;
            }

        }


        //Esta função recebe dois parâmetros: o primeiro refere-se ao destino no
qual a alocação será feita e o segundo é a chave da fonte que será alocada (fonte
dispónível).
        private static void alocarVazao(TransMassaIntervalo destino, string
keyDaOrigem) {

            Fonte origem = destino.intervalo.fontesDisponiveis[keyDaOrigem];
            FonteUsada fonteUsada = null;

            if (!destino.fontesUsadas.ContainsKey(keyDaOrigem)) {
                fonteUsada = new FonteUsada();
                fonteUsada.fonte = origem;
                destino.fontesUsadas.Add(keyDaOrigem, fonteUsada);
            } else {
                fonteUsada = destino.fontesUsadas[keyDaOrigem];
            }

            fonteUsada.vazaoUsada += calculoParaAbsorverDeltaM(destino.deltaM(),
destino.deltaMtemp(), destino.intervalo.concentracaoFinal, origem.concentracao(),
origem.vazaoDisponivel);

            //add a fonte usada ao destino
            //atualizar fonte disponivel
            origem.vazaoDisponivel -= fonteUsada.vazaoUsada;
        }

        //TransMassaIntervalosParaUmContaminante define um conjunto de deltaM de
cada intervalo e cada operação e vazão calculada
        private static void defineTransMassaIntervalosParaUmContaminante(Diagrama
diagrama) {
            String[] chaves = diagrama.operacoes.Keys.ToArray();

            for (int i = 0; i < diagrama.intervalos.Count; i++) {
                Intervalo intervalo = diagrama.intervalos[i];
                for (int j = 0; j < diagrama.operacoes.Count; j++) {
                    Operacao operacao = diagrama.operacoes[chaves[j]];

                    //Verificar se as operações estão nos intervalos. E criar uma
transMassa para cada uma delas.
```

```
                    foreach (ConcentracaoContaminanteOperacao cco in
operacao.listaConcentracoes){
                        if(diagrama.qtdContaminantes == 1 ||
cco.contaminante.nome.Equals(diagrama.contaminanteReferencia.nome)){
                            if (cco.concentracaoEntrada <=
intervalo.concentracaoInicial
                                && cco.concentracaoSaida >=
intervalo.concentracaoFinal) {
                                TransMassaIntervalo transmassa = new
TransMassaIntervalo();
                                transmassa.operacao = operacao;
                                transmassa.intervalo = intervalo;

                                intervalo.transmassas.Add(transmassa.key(),
transmassa);
                                operacao.transmassas.Add(transmassa.key(),
transmassa);
                            }
                        }
                    }
                }
            }
        }

        private static void defineIntervalos(Diagrama diagrama) {
            for (int i = 1; i < diagrama.concentracoes.Count; i++) {
                Intervalo intervalo = new Intervalo();

                intervalo.concentracaoFinal = diagrama.concentracoes[i];
                intervalo.concentracaoInicial = diagrama.concentracoes[i - 1];
                intervalo.posicao = i - 1;

                /*String[] chaves = diagrama.fontesExternas.Keys.ToArray();
                for (int j = 0; j < chaves.Length; j++) {
                    FonteExterna fe = diagrama.fontesExternas[chaves[j]];
                    Fonte fonte = new Fonte();
                    fonte.fonteExterna = fe;

                    intervalo.fontesDisponiveis.Add(chaves[j], fonte);
                }*/

                diagrama.intervalos.Add(intervalo);
            }
        }

        private static void preencheRestricoes(Diagrama diagrama, DataGridView
gridRestricoes) {
            for (int i = 0; i < gridRestricoes.RowCount; i++) {
                DataGridViewRow linha = gridRestricoes.Rows[i];
                Operacao origem =
diagrama.operacoes[linha.HeaderCell.Value.ToString()];
                for (int j = 0; j < linha.Cells.Count; j++) {
//(linha.Cells.Count = gridRestricoes.Columns.Count quando tds as colunas e
linhas estão preenchidas)
                    Operacao destino =
diagrama.operacoes[gridRestricoes.Columns[j].HeaderText];
                    bool pode = linha.Cells[j].Value.ToString().Equals("Sim");

                    Restricao restricao = new Restricao();
                    restricao.origem = origem;
                    restricao.destino = destino;
```

```
                    restricao.pode = pode;

                    diagrama.restricoes.Add(origem.nomeOperacao + "-" +
destino.nomeOperacao, restricao);
                }

            }
        }

        private static void preencheFontesExternas(Diagrama diagrama,
DataGridView gridFontesExternas) {
            for (int k = 0; k < gridFontesExternas.RowCount - 1; k++) {
                DataGridViewRow linha = gridFontesExternas.Rows[k];

                FonteExterna fe = new FonteExterna();
                fe.nome = (String)linha.Cells[0].Value;
                fe.concentracao = float.Parse(linha.Cells[1].Value.ToString());
                if (linha.Cells[2].Value == null) {
                    fe.vazao = -1;
                }
                else {
                    fe.vazao = float.Parse(linha.Cells[2].Value.ToString());
                }

                diagrama.fontesExternas.Add(fe.nome, fe);

                //Se a concentração não estiver na List do Diagrama, add a
concentração da fonte externa
                if (!diagrama.concentracoes.Contains(fe.concentracao)) {
                    diagrama.concentracoes.Add(fe.concentracao);

                }
            }

        }

        private static void preencheConcentracoesContaminantes(Diagrama diagrama,
DataGridView gridContaminantes, int qtdContaminantes, String
nomeContaminanteReferencia) {
            for (int i = 0; i < gridContaminantes.RowCount; i++) {
                DataGridViewRow linha = gridContaminantes.Rows[i];

                Contaminante contaminante = new Contaminante();
                contaminante.nome = linha.Cells[1].Value.ToString();
                if (contaminante.nome.Equals(nomeContaminanteReferencia)) {
                    contaminante.referencia = true;
                    diagrama.contaminanteReferencia = contaminante;
                }

                if (!diagrama.contaminantes.ContainsKey(contaminante.nome)) {
                    diagrama.contaminantes.Add(contaminante.nome, contaminante);
                }
                else {
                    contaminante = diagrama.contaminantes[contaminante.nome];
                }

                ConcentracaoContaminanteOperacao cco = new
ConcentracaoContaminanteOperacao();
                cco.concentracaoEntrada =
float.Parse(linha.Cells[2].Value.ToString ());
                cco.concentracaoSaida =
float.Parse(linha.Cells[3].Value.ToString());
```

```
                cco.contaminante = contaminante;
                cco.operacao =
diagrama.operacoes[linha.Cells[0].Value.ToString()];
                cco.contaminante.listaConcentracoes.Add(cco);
                cco.operacao.listaConcentracoes.Add(cco);


                //Se a concentração não estiver na List do Diagrama, add as
concentrações de entrada e de saída dos contaminantes
                if (!diagrama.concentracoes.Contains(cco.concentracaoEntrada)) {
                    diagrama.concentracoes.Add(cco.concentracaoEntrada);
                }

                if (!diagrama.concentracoes.Contains(cco.concentracaoSaida)) {
                    diagrama.concentracoes.Add(cco.concentracaoSaida);
                }
            }

            //O bloco abaixo é executado para o contaminante de referência
            if (qtdContaminantes > 1) {
                //funçaoAjusteConcentracoes
                //funcaoAtualizarGridContaminantes

            }
        }

        private static void preencheOperacoes(Diagrama diagrama, DataGridView
dgvOperacoes, int qtdContaminante, String nomeOperacaoReferencia) {
            for (int k = 0; k < dgvOperacoes.RowCount -1; k++) {
                DataGridViewRow linha = dgvOperacoes.Rows[k];

                Operacao op = new Operacao();
                op.nomeOperacao = (String)linha.Cells[0].Value;
                op.fk = float.Parse(linha.Cells[1].Value.ToString());
                op.id = k;

                if (qtdContaminante > 1 &&
op.nomeOperacao.Equals(nomeOperacaoReferencia)) {
                    op.referencia = true;
                    diagrama.operacaoReferencia = op;
                }

                diagrama.operacoes.Add(op.nomeOperacao, op);
            }
        }
    }


}
```

Arquivo Diagrama.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DFA_Teste0.entidades {
    public class Diagrama {
        public Dictionary<String, Operacao> operacoes { get; private set; }
```

```
        public Dictionary<String, Contaminante> contaminantes { get; private set;
}
        public Dictionary<String, FonteExterna> fontesExternas { get; private
set; }
        public Dictionary<String, Restricao> restricoes { get; private set; }
        public List<float> concentracoes { get; private set; }
        public List<Intervalo> intervalos { get; private set; }
        public Contaminante contaminanteReferencia { get; set; }
        public Operacao operacaoReferencia { get; set; }
        public int qtdContaminantes { get; set; }

        //Criar os construtores
        public Diagrama() {
            this.operacoes = new Dictionary<String, Operacao>();
            this.contaminantes = new Dictionary<String, Contaminante>();
            this.fontesExternas = new Dictionary<String, FonteExterna>();
            this.restricoes = new Dictionary<String, Restricao>();
            this.concentracoes = new List<float>();
            this.intervalos = new List<Intervalo>();

        }

        public float totalFontesAlocadasNoIntervalo(int pos) {
            Intervalo intervalo = this.intervalos[pos];
            float total = 0;
            float concentracaoMinima = intervalo.concentracaoInicial;

            foreach (TransMassaIntervalo transMassa in
intervalo.transmassas.Values) {
                total += transMassa.vazaoParaDeltaM();
            }

            for (int i = pos + 1; i < this.intervalos.Count; i++) {
                for (int j = 0; j <
this.intervalos[i].fontesExternasUsadas.Values.Count; j++) {
                    FonteUsada fonte =
this.intervalos[i].fontesExternasUsadas.Values.ToArray()[j];
                    if (fonte.fonte.concentracao() <= concentracaoMinima){
                        total += fonte.vazaoUsada;
                    }

                }
                foreach (TransMassaIntervalo transMassa in
this.intervalos[i].transmassas.Values) {
                    foreach(FonteUsada fonteUsada in
transMassa.fontesUsadas.Values) {
                        if (fonteUsada.fonte.fonteInterna != null) {
                            if (fonteUsada.fonte.fonteInterna.concentracao <=
concentracaoMinima) {
                                total += fonteUsada.vazaoUsada;
                            }
                        }
                    }
                }

            }

            return total;

        }
        public float constanteTransferenciaLinear(Contaminante contaminante,
Operacao operacao) {
```

```
            float resultado = 0;
            float deltaCj = 0;
            float deltaCref = 0;

            for (int i = 0; i < operacao.listaConcentracoes.Count; i++) {
                if
(operacao.listaConcentracoes[i].contaminante.nome.Equals(contaminante.nome)) {
                    deltaCj = operacao.listaConcentracoes[i].concentracaoSaida -
operacao.listaConcentracoes[i].concentracaoEntrada;
                }

                if (this.contaminanteReferencia == null) {
                    deltaCref = deltaCj;
                } else if
(operacao.listaConcentracoes[i].contaminante.nome.Equals(this.contaminanteReferen
cia.nome)) {
                    deltaCref = operacao.listaConcentracoes[i].concentracaoSaida
- operacao.listaConcentracoes[i].concentracaoEntrada;
                }
            }

            resultado = deltaCj / deltaCref;

            return resultado;
        }

        public float totalFonteExternaUsada(string nomeFonteExterna) {
            float resultado = 0;
            foreach (Intervalo intervalo in this.intervalos) {
                foreach (TransMassaIntervalo transMassa in
intervalo.transmassas.Values) {

                    foreach (FonteUsada fonteUsada in
transMassa.fontesUsadas.Values) {
                        if (fonteUsada.fonte.fonteExterna != null &&
fonteUsada.fonte.fonteExterna.nome == nomeFonteExterna) {
                            resultado += fonteUsada.vazaoUsada;
                        }

                    }

                }

            }
            return resultado;
        }
        public float menorConcentracaoFonteExterna() {
            if (this.fontesExternas.Values.Count == 0) {
                throw new Exception("Preencha pelo menos uma fonte externa");
            }

            float resultado = fontesExternas.Values.First().concentracao;

            for (int i = 0; i < this.fontesExternas.Values.Count; i++) {
                FonteExterna fe = this.fontesExternas.Values.ElementAt(i);
                if (fe.concentracao < resultado) {
                    resultado = fe.concentracao;
                }

            }
            return resultado;
        }
```

```
        //Cálculo das concentrações de cada contaminante de cada operação após o
WSD. Se houver violações, elas serão mantidas.

        public float concentracaoFinal(Operacao op, Contaminante contaminante) {

            float k = constanteTransferenciaLinear(contaminante, op);
            float deltaCref = op.concentracaoFinalAjustada() -
op.concentracaoInicialAjustada();
            float ci = concentracaoInicial(op, contaminante);

            return ((k*deltaCref) + ci);
        }

        public float concentracaoInicial(Operacao op, Contaminante contaminante)
{
            float somatorio = 0;
            List<FonteUsada> fontesUsadas = op.fontesUsadasNaOperacao();
            for (int i = 0; i < fontesUsadas.Count; i++) {
                FonteUsada fu = fontesUsadas[i];

                float vazao = fu.vazaoUsada;
                float concentracao = 0;
                if (fu.fonte.fonteInterna != null) {
                    concentracao =
concentracaoFinal(fu.fonte.fonteInterna.transMassa.operacao, contaminante);
                }
                else {
                    concentracao = fu.fonte.fonteExterna.concentracao;
                }

                somatorio += vazao * concentracao;

            }

            return somatorio / op.vazaoUsada();
        }


    }


}
```