

2

Construção de vocabulário

A partir de pesquisas sobre a heurística busca tabu, Glover [11] propôs a técnica de construção de vocabulário. Essa técnica está relacionada com heurísticas que visam a criação de novas soluções em função das características de outras, como em análise alvo [11] e reconexão por caminhos [14]. Segundo Glover [11], em análise alvo pode-se implicitamente combinar ou subdividir atributos para melhorar a decisão de um movimento na busca tabu. Já em reconexão por caminhos, buscam-se novas soluções explorando trajetórias no espaço de soluções que conectam soluções de elite.

A idéia básica de construção de vocabulário consiste em gerar um conjunto de fragmentos de soluções e recombiná-los até obterem-se soluções completas [12]. Identificam-se fragmentos significativos das soluções, ao invés de só focar em soluções completas. O conjunto desses fragmentos é progressivamente enriquecido e combinado para criar fragmentos maiores, até finalmente produzirem soluções completas.

Este processo recebe seu nome por analogia com o processo de construção de vocabulário, em que há o enriquecimento do conjunto de palavras conhecidas sempre que for necessário representar um novo conhecimento. Construções significativas em cada nível podem ser visualizadas como representações de “palavras de ordem superior”, assim como em linguagem natural geram-se novas palavras para substituir coleções de palavras que personificam conceitos úteis [15]. As palavras contêm características das soluções, que podem representar uma solução completa ou apenas fragmentos delas. Um conjunto de palavras, através de alguma combinação, forma uma frase. Uma sentença é uma frase que representa uma solução completa. Na analogia com a técnica de construção de vocabulário, combinar fragmentos de soluções até obter soluções completas é como combinar palavras em frases até formar-se uma sentença completa.

O motivo subjacente à construção de vocabulário é obter vantagem dos contextos onde certas configurações parciais de soluções ocorrem freqüentemente como componentes de boas soluções completas. Além disso, uma estratégia de procurar “boas configurações parciais” (bons elementos de voca-

bulário) e tratá-las como uma unidade pode ajudar a contornar a explosão combinatorial, que potencialmente resulta da simples manipulação dos elementos mais primitivos por si só.

Em uma classificação mais recente [12], a técnica de construção de vocabulário pode ser vista como um procedimento de programação com memória adaptativa, relacionada com métodos baseados em reconexão por caminhos e algoritmos evolucionários. Ainda segundo Glover [14], a construção de vocabulário pode ser qualificada como uma instância de reconexão por caminhos. Há dois objetivos principais: identificar uma boa coleção de pontos de referência e identificar caminhos em um ou mais espaços de vizinhança que unirão componentes destas soluções parciais, com modificações adequadas, para produzir soluções completas.

O diferencial desta técnica é como as soluções são consideradas. Na visão de construção de vocabulário, analisar um conjunto de soluções é tratá-lo como um texto, de modo a descobrir combinações de características compartilhadas pelas soluções. As combinações de características, que são significantes o bastante, são qualificadas como unidade de vocabulário e são tratadas como novos atributos. As unidades de vocabulários podem ser combinadas em unidades maiores, como uma base para a construção de novas soluções.

2.1 Descrição geral

Uma visão geral do processo de construção de vocabulário está ilustrada na Figura 2.1 [15].

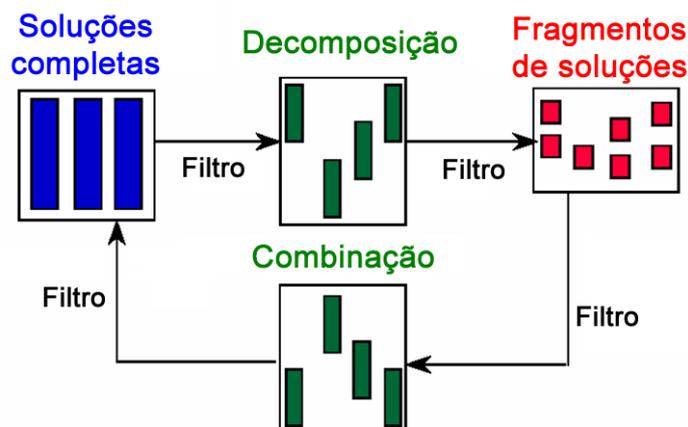


Figura 2.1: Processo de construção de vocabulário

Para exemplificar o processo de construção de vocabulário é apresentada a estratégia definida por Glover [11, 14]. As operações para encontrar palavras e formar frases, apresentadas a seguir, são simples e aplicáveis a diversos problemas combinatórios.

Um atributo é representado em uma palavra através de uma atribuição $y_j = p$, que deve ser lida como o atributo j da palavra y possuindo valor p . Uma palavra é composta por diversos atributos. Pelo fato de sua dimensão e codificação poderem ser diferentes das soluções do problema, justifica-se diferenciar uma solução de uma palavra. Uma solução pode ser mapeada para uma palavra e vice-versa.

Sejam \mathcal{U} o conjunto de atributos e y' e y'' duas palavras. Define-se o operador de *interseção* pela regra:

$$INT(y', y'') = z$$

$$z_u = \begin{cases} y'_u, & \text{se } y'_u = y''_u \\ *, & \text{se } y'_u \neq y''_u, \end{cases}$$

aplicada a cada atributo $u \in \mathcal{U}$.

Para avaliar a qualidade das palavras são definidas algumas medidas. A função $Size(y)$ indica a quantidade de atributos de y que são diferentes de “*”, ou seja, o tamanho de uma palavra. Dadas duas palavras y e x , o operador \subseteq é definido de forma que $y \subseteq x$ implica que $y_u = x_u$ ou $y_u = *$, para todo $u \in \mathcal{U}$. Com este operador, define-se o conjunto:

$$Enclosure(y, \mathcal{Y}) = \{x \in \mathcal{Y} \mid y \subseteq x\}.$$

Deste modo, $EncValue(y, \mathcal{Y}) = |Enclosure(y, \mathcal{Y})|$ indica a quantidade de palavras do conjunto \mathcal{Y} que contêm a palavra y .

A partir dessas definições, Glover sugere duas abordagens para a construção de um método para buscar palavras em um conjunto de soluções. Vale lembrar que as soluções são mapeadas em palavras e a diferença entre palavra e solução está no fato de uma palavra poder representar uma solução incompleta. Uma palavra é obtida pela sucessiva aplicação do operador de interseção as soluções obtidas de um conjunto de soluções elite \mathcal{X} . A primeira abordagem prima por obter palavras que possuam o maior valor possível para $EncValue(y, \mathcal{X})$, respeitando $Size(y) \geq s_{min}$, onde s_{min} é um parâmetro que estabelece o menor tamanho de uma palavra. Ou seja, uma palavra obtida a partir do máximo possível de interseções de soluções cujo tamanho seja maior do que um dado valor mínimo. A outra abordagem busca palavras, por exemplo a palavra y , que possuam o maior valor possível para $Size(y)$, respeitando

a restrição $EncValue(y, \mathcal{X}) \geq v_{min}$, onde v_{min} é um parâmetro que estabelece a menor quantidade de soluções utilizadas para se obter a palavra y .

A solução de um problema de otimização combinatória pode ser frequentemente representada por uma combinação ou uma permutação de seus elementos. Por exemplo, no problema da mochila um elemento x_i pode pertencer à solução ($x_i = 1$) ou não ($x_i = 0$), podendo esta relação de pertinência ser um atributo da solução. No problema do caixeiro viajante, um atributo interessante é a precedência dos elementos na solução, onde $x_{ij} = 1$ se na solução do problema a cidade i for visitada antes de j ; $x_{ij} = 0$ caso contrário. Os problemas de seqüenciamento podem ser representados dessa última forma.

A seguir é exemplificado o funcionamento do operador INT sobre um conjunto de soluções que foram mapeadas em palavras. Determina-se que uma palavra é válida se seu tamanho for maior que um dado limite. Dado o conjunto de palavras apresentado na Tabela 2.1, uma combinação da aplicação do operador INT é apresentado na Tabela 2.2.

y_1	0	0	1	1	0	0	0	0	1	0
y_2	1	0	1	1	0	0	0	0	0	0
y_3	0	0	1	1	1	0	0	0	0	0
y_4	1	0	0	1	1	0	0	0	0	0
y_5	0	0	1	0	0	0	0	1	1	0

Tabela 2.1: Conjunto de soluções

$INT(y_1, y_4)$	*	0	*	1	*	0	0	0	*	0
$INT(INT(y_1, y_4), y_3)$	*	0	*	1	*	0	0	0	*	0
$INT(INT(INT(y_1, y_4), y_3), y_5)$	*	0	*	*	*	0	0	*	*	0

Tabela 2.2: Utilização do operador INT

Para exemplificar o que este resultado pode significar, suponha que este conjunto represente soluções do problema da mochila. Cada posição no vetor representa a pertinência dos objetos, sendo “1” para presente, “0” para ausente e “*” para não se sabe. A palavra obtida $[*, 0, *, *, *, 0, 0, *, *, 0]$ identifica elementos que não pertencem a boas soluções. A partir deste novo conhecimento, pode-se definir um novo problema onde há um conjunto reduzido de elementos a serem escolhidos.

Um exemplo onde o tamanho mínimo de uma palavra válida é seis e a seqüência utilizada para compor as palavras é y_2, y_3, y_4, y_5, y_1 está apresentado na Tabela 2.3.

A palavra w_3 não é válida, pois possui tamanho quatro, que é menor do que o valor definido para uma palavra válida. Esta restrição é útil para evitar que se trabalhe com fragmentos de solução muito pequenos.

$w_1 = INT(y_2, y_3)$	*	0	1	1	*	0	0	0	0	0
$w_2 = INT(w_1, y_4)$	*	0	*	1	*	0	0	0	0	0
$w_3 = INT(w_2, y_5)$	*	0	*	*	*	0	0	*	*	0
$w_4 = INT(w_2, y_1)$	*	0	*	1	*	0	0	0	1	0

Tabela 2.3: Exemplo da aplicação do operador INT

Dado um conjunto de palavras, essas podem ser combinadas até formarem frases completas através do operador de *interseção estendido*. As frases completas são chamadas de sentenças. Sejam \mathcal{U} o conjunto de atributos e y' e y'' duas palavras. Define-se o operador de *interseção estendido* pela regra:

$$EINT(y', y'') = z$$

$$z_u = \begin{cases} y'_u, & \text{se } y''_u = * \\ y''_u, & \text{se } y'_u = * \\ y'_u, & \text{se } y'_u = y''_u \\ \#, & \text{se } y'_u \neq y''_u, \end{cases}$$

aplicada a cada atributo $u \in \mathcal{U}$.

Para exemplificar o uso do operador $EINT$, considere o seguinte conjunto de palavras representado na Tabela 2.4. Um exemplo da combinação das palavras y_1 , y_5 e y_2 está apresentado na Tabela 2.5.

y_1	0	*	1	1	0	*	0	*	*	0
y_2	*	0	1	1	*	0	*	0	*	0
y_3	0	*	1	*	*	0	*	*	0	*
y_4	1	0	*	1	*	*	0	0	0	0
y_5	*	*	1	*	0	*	*	1	1	*

Tabela 2.4: Conjunto de palavras

$EINT(y_1, y_5)$	0	*	1	1	0	*	0	1	1	0
$EINT(EINT(y_1, y_5), y_2)$	0	0	1	1	0	0	0	0	1	0

Tabela 2.5: Utilização do operador $EINT$

Um agregado de palavras formado a partir de sucessivas aplicações do operador $EINT$ recebe o nome de frase. A frase $EINT(y_1, y_5)$ é chamada de frase incompleta enquanto $EINT(EINT(y_1, y_5), y_2)$ de sentença (ou frase completa).

Frases podem ter valores inconsistentes ('#') que podem ser considerados como inconsistências sintáticas. Um exemplo de frase com valores inconsistentes é apresentado na Tabela 2.6.

$EINT(y_1, y_3)$	0	*	1	1	0	0	0	*	0	0
$EINT(EINT(y_1, y_3), y_4)$	#	0	1	1	0	0	0	0	0	0

Tabela 2.6: Utilização do operado $EINT$ gerando frase inconsistente

Além dessas inconsistências sintáticas, ainda podem ocorrer inconsistências semânticas. Por exemplo, no problema da mochila pode-se obter uma sentença completa que represente uma solução inviável em relação às restrições do problema. Apesar disso, assim como frases incompletas, as sentenças inconsistentes também contêm informações significativas sobre soluções para o problema.

Uma sentença que não possui qualquer tipo de inconsistência é dita significativa. As sentenças significativas são soluções viáveis para o problemas. Eventualmente, elas ainda podem ser aperfeiçoadas por outros métodos.

2.2

Revisão da literatura

Apesar da técnica de construção de vocabulário ainda ser pouco estudada, é possível encontrar na literatura algumas de suas aplicações.

Um dos trabalhos pioneiros em construção de vocabulário foi o de Rochat e Taillard [28]. A técnica foi aplicada na resolução do problema de roteamento de veículos (VRP, do inglês *Vehicle Routing Problem*), em sua versão elementar (CVRP) e na variante com janelas de tempo (CVRPTW). O problema elementar de roteamento de veículos idênticos consiste em determinar rotas para veículos que partem de um único depósito e devem atender uma seqüência de consumidores de um conjunto $C = \{1, \dots, n\}$, onde a distância entre os consumidores i e j é c_{ij} , entregando quantidades de mercadoria q_i a cada consumidor $i = 1, \dots, n$, de modo que:

- seja minimizada a distância percorrida pelos veículos;
- somente um veículo visite cada consumidor; e
- seja respeitado o limite Q de carga de mercadorias de cada veículo.

No problema de roteamento de veículos com janelas de tempo, as distâncias entre os consumidores c_{ij} são vistas como o tempo gasto de ir de i até j e cada consumidor i requer um tempo de serviço t_i . A visita ao consumidor i deve se realizar durante uma janela de tempo $[b_i, e_i]$ pré-estabelecida.

Uma das perspectivas que foram utilizadas no desenvolvimento do método foi determinar quais são as variáveis consistentes nas boas soluções encontradas durante a busca. Segundo os autores, uma variável consistente é aquela que se determina fortemente e freqüentemente em um valor específico

(ou em um intervalo curto). De um modo mais específico, quanto mais frequentemente a variável receber um determinado valor nas melhores soluções, mais altamente se qualifica como consistente. Essa é a idéia de se identificar fragmentos significativos nas boas soluções.

O método proposto pelos autores possui duas etapas na sua execução. A geração de boas soluções e a criação do conjunto de fragmentos são os objetivos da primeira fase. Fragmentos de uma solução são definidos como as rotas de cada veículo. À medida que são geradas novas soluções, o conjunto de rotas é ampliado com as novas rotas encontradas. Espera-se que toda a informação necessária para criar novas soluções de boa qualidade exista nessas soluções iniciais, mas de um modo não aparente. No caso do VRP, essa informação está incluída nas rotas. Então, espera-se que a primeira fase crie um conjunto de rotas que inclua membros não muito diferentes dos que são encontradas em boas soluções. Não é irreal pensar que seja fácil criar boas rotas. O desafio é encontrar um conjunto de rotas cujos elementos sejam simultaneamente bons para todos os consumidores.

Na segunda etapa, a meta é extrair as rotas boas e melhorá-las. O autor parte do princípio que, se uma nova solução possui rotas que pertençam a uma boa solução conhecida, então o valor da função objetivo da nova solução é, provavelmente, melhor do que o de outra solução que não contenha essas rotas. Portanto, a segunda fase está relacionada com a extração de rotas de modo a intensificar a busca em novas soluções que contenham essas rotas.

Foi proposta uma técnica de pós-otimização, em que se tenta obter a melhor combinação das rotas, de modo a formar uma solução completa válida. Isso é feito através de um modelo de programação inteira. Segundo o autor, essa abordagem torna o problema mais difícil de ser resolvido, além de não acrescentar novas informações para o conjunto de rotas. O propósito dessa proposta foi obter soluções relaxadas, de tal forma que a solução pudesse ser viabilizada e melhorada por uma busca local. Nem sempre é possível reconstruir soluções completas a partir dos fragmentos conhecidos. Contudo, ainda é desejável soluções incompletas que possuam variáveis consistentes, pois estas podem ser convertidas em soluções completas e viáveis.

Esta é uma clara aplicação da idéia de construção de vocabulário. Geram-se boas soluções, a partir das quais encontram-se palavras. Essas palavras são combinadas em frases e, por fim, conseguem-se montar novas soluções.

Os resultados desse trabalho são expressivos, encontrando as melhores soluções conhecidas para diversas instâncias teste do VRP.

Na mesma época, Kelly e Xu [19] propuseram uma heurística baseada em particionamento de conjuntos para a resolução do VRP. Os autores

exploraram modos de usar uma geração simples com refinamento heurístico para construção de soluções, que sejam competitivas ou melhores do que as obtidas pelos algoritmos mais complicados.

Considera-se que uma solução pobre, produzida por uma heurística simples, pode conter um subconjunto de “boas” rotas. Baseado nisto, foi proposto um procedimento de duas fases como solução do VRP. Na primeira fase, heurísticas simples são utilizadas para gerar um número suficiente de rotas diferentes. Na segunda fase, identificam-se as “boas” rotas que são combinadas para formar uma solução através de uma heurística baseada em particionamento de conjuntos. O modelo de particionamento de conjuntos é utilizado para realizar a tarefa de consolidação na segunda fase. A heurística proposta requer o uso de técnicas sofisticadas de busca, pois o problema de particionamento de conjuntos (SPP, do inglês *set-partitioning problem*) é um problema NP-difícil. Para tal, foi desenvolvida uma heurística de busca tabu para solucioná-lo. Como heurísticas simples podem não produzir todas as rotas das melhores soluções e o modelo de particionamento de conjunto é resolvido de modo aproximado, aprimorou-se o processo criando novas rotas durante o processo de busca.

A fase de geração de rotas utiliza-se de heurísticas clássicas para o VRP. Após N rotas serem produzidas, emprega-se o modelo de particionamento de conjuntos para selecionar m rotas e montá-las em uma solução. A formulação matemática utilizada para a solução do problema é apresentada a seguir. Seja c_j o custo (nesse caso, o comprimento da rota) de uma rota j , e a_{ij} um valor binário igual a 1 se, e somente se, a rota j contém o consumidor i ; e $a_{ij} = 0$ caso contrário. Define-se x_j como uma variável binária tal que $x_j = 1$ se, e somente se, a rota j for selecionada para a solução; $x_j = 0$ caso contrário. O modelo de programação inteira é formulado como:

$$\text{minimizar } \sum_{j=1}^N c_j x_j \quad (2-1)$$

$$\text{sujeito a: } \sum_{j=1}^N a_{ij} x_j = 1, \quad i = 1, \dots, n, \quad (2-2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, N. \quad (2-3)$$

Nesta formulação, a função objetivo consiste em minimizar a soma dos comprimentos das rotas. A restrição (2-2) especifica que cada consumidor deve ser coberto pela solução somente uma única vez (essa restrição é relaxada na heurística utilizada, para permitir mais combinações).

Segundo os autores, a abordagem defendida por eles tem muitas similaridades com o método proposto por Rochat e Taillard [28]. Ambos os

métodos procuram combinar inteligentemente rotas geradas previamente por outro método. No trabalho apresentado pelos autores, é utilizado um conjunto enorme de rotas iniciais e as soluções são produzidas por um procedimento elaborado de atualização e recombinação de rotas, enquanto Rochat e Taillard usam um conjunto reduzido de rotas iniciais e criam novas soluções usando seleção probabilística combinada com um procedimento de melhoria local. Não obstante, esse método de busca local é um dos algoritmos mais eficientes conhecidos da literatura e também um resolvidor comercial de programação inteira (CPLEX) é utilizado para resolver o modelo de particionamento como pós-otimização.

Um método baseado em busca tabu para resolver o problema de escalonamento da produção de tiras quentes de aço foi proposto por Lopez, Carter e Gendreau [21]. Este problema é baseado em um estudo de caso real para uma indústria canadense, com propósito de aplicação prática.

Segundo a análise desse problema pelos autores como um problema de programação matemática, há duas atividades importantes associadas. A primeira é relacionada com os pedidos que devem ser escolhidos para produção (seleção) e a segunda com o seqüenciamento dos produtos listados nos pedidos selecionados (seqüenciamento). A seleção dos pedidos pode ser representada como um problema da mochila. Determinar a seqüência de produção dos produtos selecionados corresponde ao bem conhecido problema do caixeiro viajante assimétrico. Para uma modelagem completa do problema, é apresentada uma formulação matemática através de uma generalização do caixeiro viajante com coleta de prêmios. Como são problemas de difícil solução exata, optou-se por uma heurística baseada em busca tabu.

O algoritmo é composto de duas fases. A primeira gera um conjunto de soluções iniciais. A segunda fase, chamada de canibalismo, utiliza esse conjunto para melhorar outras soluções. O canibalismo consiste em identificar blocos ruins na solução corrente e trocá-los por blocos bons de outras soluções, atendendo melhor aos requisitos do problema. Vale ressaltar a idéia básica utilizada: gerar diversas soluções de modo a aproveitar fragmentos significativos para se produzir novas soluções.

O problema de escalonar a produção de um produto que tem vários modelos em uma linha de montagem flexível foi tratado por Scholl, Klein e Domschke [30]. Para resolver esse problema foi desenvolvido um procedimento de busca tabu com uma estratégia de construção de vocabulário baseada em padrões. Unidades de vocabulário são definidas através de seqüências parciais de modelos dos produtos. Uma seqüência parcial é chamada padrão se a linha de produção retorna para seu estado inicial após processar os produtos da

seqüência. Um padrão é dito elementar se não existe uma decomposição em padrões menores. Esses padrões são gerados e escolhidos por um procedimento de geração de colunas e regras heurísticas de transformação, baseados numa reformulação do problema original.

Foi então definido o problema de combinação de padrões (PCP), que consiste em combinar padrões elementares para atender as demandas do problema original. Ele pode ser modelado como um problema de programação linear inteira. Se todos os padrões elementares necessários são conhecidos, a solução desse problema é solução para o problema original. Entretanto, essa abordagem se depara com a dificuldade principal do número de padrões elementares ser usualmente muito grande. Além do mais, é custoso identificar e excluir padrões não elementares a serem considerados na resolução do problema.

Desenvolveu-se uma abordagem mais apropriada baseada em construção de vocabulário, que permitiu produzir e combinar somente um subconjunto significativo de padrões. Primeiramente, é construído um vocabulário “difuso” que necessita ser refinado em fases subseqüentes. Um subconjunto de padrões é determinado pela solução da relaxação linear do PCP com um procedimento de geração de colunas. Embora as quantidades com que cada padrão é escolhido possam não ser inteiras, a solução do problema relaxado é transformada em uma solução do PCP através de um procedimento heurístico de arredondamento. Entretanto, esse procedimento não garante que todos os padrões elementares sejam produzidos. Para contornar esse inconveniente, a aplicação de uma busca tabu, que utiliza informações codificadas nos padrões encontrados anteriormente, permite aprimorar a seqüência de padrões que representa a solução para o problema original. Em comparação com os outros métodos conhecidos da literatura, a versão da busca tabu que utiliza a informação codificada no vocabulário domina as outras abordagens com respeito à qualidade da solução e velocidade computacional.

O procedimento de construção de vocabulário baseado em padrões apresentado é altamente influenciado pelo comprimento máximo do padrão. Enquanto o tempo computacional requerido para gerar o vocabulário cresce exponencialmente com um aumento do valor deste parâmetro, a complexidade do problema de seqüenciamento dos padrões diminui, e vice versa. Se o comprimento do padrão é grande, as decisões de arredondamento na viabilização de uma solução relaxada do PCP tendem a ser críticas com relação ao valor da função objetivo. Em contrapartida, os efeitos compensatórios relacionados com a utilização dos padrões podem não ser obtidos se o comprimento máximo do padrão for muito pequeno. Por exemplo, perde-se tempo considerando um

grande volume de unidades de vocabulário pouco significativas ao invés de aproveitar o conhecimento extraído de blocos maiores e mais significativos.

Um novo paradigma para otimização combinatória foi proposto por Toulouse, Thulasiraman e Glover [33], chamado de *Multi-Level Cooperative Search*. A proposta desse paradigma é uma estratégia para organizar, de um modo configurável, a rede de dependências entre programas cooperativos. Algoritmos cooperativos necessitam de mecanismos para compartilhar informação e melhorar o comportamento da busca em programas paralelos. A estratégia proposta usa princípios advindos de *scatter search* e construção de vocabulário para guiar a “combinação” da informação compartilhada e originada nas diferentes fontes. Assume-se que as soluções de problemas de otimização combinatória possam ser representadas como seqüências ou partições. Variáveis de decisão, que definem o valor da função objetivo, servem como uma escolha lógica para constituir os componentes de mais baixo nível, de uma estrutura de vários níveis. Progressivamente, combinam-se estas variáveis para prover maiores agregados de variáveis. Forma-se uma estrutura em diversos níveis, onde níveis superiores trabalham com agregados maiores que compõem as soluções.

Combs [6] propôs uma busca tabu adaptativa para solucionar o problema de escalonamento da tripulação de reabastecedores aéreos. O problema consiste em se determinar a tripulação que fará um vôo de reabastecimento de uma série de aeronaves, sujeito a diversas restrições. Por exemplo, cada aeronave que será reabastecida é atendida por apenas uma tripulação e após uma série de vôos as tripulações devem terminar seu turno na base de onde partiram. Existem diversas bases de onde partem as tripulações. Uma das abordagens desenvolvidas para solucionar o problema combina busca tabu adaptativa com o problema de particionamento de conjuntos. O sistema desenvolvido foi implementado em OpenTS, um *framework* Java para busca tabu [18]. Define-se como um segmento de vôo qualquer vôo sem parada entre pares de bases operacionais. Existem diversos pontos para reabastecimento entre a partida e a chegada nas bases operacionais. A partir dessa definição, o autor formulou um problema baseado no SPP seguindo a abordagem de Kelly e Xu [19]. A resolução do SSP foi feita pelo CPLEX, mas, muitas vezes, após várias horas de computação, não obteve-se o valor ótimo ou, pior, se esgotou a memória do resolvidor. Por isso, também foi desenvolvida uma heurística que a partir de soluções incompletas do SPP permite reconstruir e melhorar as soluções geradas na construção de vocabulário. Essa abordagem, baseada em construção de vocabulário, foi o último melhoramento realizado na busca tabu e possibilitou um aumento expressivo na robustez do método proposto.

Uma aplicação de técnicas de mineração de dados no contexto de oti-

mização foi apresentada por Ribeiro, Plastino e Martins [25]. Mineração de dados faz referência à extração de conhecimento novo e potencialmente útil a partir de um conjunto de dados em termos de padrões e regras. Os autores acreditam que técnicas de mineração de dados podem ser utilizadas para extrair padrões que representam características de soluções sub-ótimas de um problema de otimização combinatorial. Além disso, esses padrões podem ser utilizados para guiar a busca para soluções melhores em procedimentos metaheurísticos.

Foi proposta uma hibridização da metaheurística GRASP [7] que incorpora o processo de mineração de dados. Basicamente, após a heurística GRASP executar um número significativo de iterações, o processo de mineração de dados extrai padrões de um conjunto de soluções de elite que guia as iterações GRASP seguintes. Soluções são representadas por conjuntos de itens. Padrões são definidos como subconjuntos de itens que ocorrem num número significativo de soluções. O processo de obter estes padrões é o bem conhecido problema em mineração de dados chamado Mineração de Conjunto Freqüente de Itens (FIM, do inglês *Frequent Itemset Mining*).

Foram apresentadas estratégias distintas pela variação de como se gera e utiliza o conjunto de padrões. Essas estratégias diferentes não foram propostas de uma vez, mas como uma evolução da pesquisa. Por exemplo, a primeira é chamada maior padrão híbrido. Ela utiliza apenas o maior padrão encontrado num conjunto de padrões. Esse conjunto é formado pelos padrões presentes em pelo menos um percentual de elementos do conjunto de soluções elite. A segunda estratégia, chamada maior suporte híbrido, procura pelo padrão mais freqüente no conjunto elite dado um tamanho mínimo. Assim, foram variando o modo de utilização dos algoritmos na mineração de dados para obter padrões diferentes. Os padrões identificados são utilizados como ponto de partida para a fase de construção GRASP e, posteriormente, refinados na busca local.

Seguindo a proposta de Glover [14], Fernandes [8] utilizou a técnica de construção de vocabulário em suas heurísticas para resolução do problema de seqüenciamento de DNA por hibridação, cuja solução é uma seqüência de DNA formada a partir de um encadeamento de subseqüências. Essa seqüência pode ser vista como um conjunto de arestas que ligam vértices representando subseqüências de DNA. O conjunto de arestas é representado por um vetor de adjacência. A partir de um conjunto de soluções de elite, as soluções são mapeadas em vetores de adjacência onde se utiliza os operadores *INT* e *EINT* para encontrar e combinar palavras. Entretanto, as soluções produzidas pelo método podem conter algumas inconsistências lógicas como, por exemplo, várias arestas direcionadas para uma mesma subseqüência de DNA. Para

contornar essa dificuldade, cada solução produzida pelo método é analisada e, se necessário, é aplicado um algoritmo de viabilização das soluções. Os resultados obtidos demonstram os benefícios da utilização da técnica como complementar às outras heurísticas utilizadas.

Na mesma linha, Aloise [1] agregou construção de vocabulário às suas heurísticas para o projeto de redes com funções de custos discretas. Cada solução desse problema é uma atribuição de valores para as arestas de uma rede, de modo a respeitar as restrições de largura de banda e demanda dos nós, entre outras. Uma representação natural é um vetor em que cada posição representa o valor alocado para uma aresta. A partir de um conjunto de soluções de elite, encontram-se palavras que por sua vez são combinadas em frases, conforme a estratégia proposta por Glover. Observou-se que o método produz muitas frases incompletas, embora haja informação relevante sobre a solução ótima. Para aproveitar estas frases, utiliza-se um modelo de programação inteira onde os valores encontrados são fixados e apenas as lacunas são determinadas pelo resolvidor. A técnica foi utilizada de modo complementar para compor a estratégia, juntamente com outras heurísticas propostas.