Marcos Daniel V. Baroni

A Hybrid Heuristic for the Multi-objective Knapsack Problem

Vitória - Espírito Santo - Brasil
November 27, 2017

Marcos Daniel V. Baroni

A Hybrid Heuristic for the Multi-objective Knapsack Problem

Tese de Doutorado apresentada de acordo com o regimento do Programa de Pós-graduação em Informática da Universidade Federal do Espírito Santo.

Universidade Feredal do Espírito Santo – UFES
Departamento de Informática
Programa de Pós-Graduação em Informática

Advisor: Dr. Flávio Miguel Varejão

Vitória - Espírito Santo - Brasil
November 27, 2017

Marcos Daniel V. Baroni

A Hybrid Heuristic for the Multi-objective Knapsack Problem

Tese de Doutorado apresentada de acordo com o regimento do Programa de Pós-graduação em Informática da Universidade Federal do Espírito Santo.

Work approved. Vitória - Espírito Santo - Brasil, November 27, 2017:

Dr. Flávio Miguel Varejão
Advisor

Dr.ª Maria Claudia Silva Boeres
Member

Dr. Arlindo Gomes de Alvarenga
Guest

Dr.ª Simone de Lima Martins
Guest

Vitória - Espírito Santo - Brasil
November 27, 2017

Resumo

Many real applications like project selection, capital budgeting and cutting stock involves optimizing multiple objectives that are usually conflicting and can be modelled as a multi-objective knapsack problem (MOKP). Unlike the single-objective case, the MOKP is considered a NP-Hard problem with considerable intractability. This work propose a hybrid heuristic for the MOKP based on the shuffled complex evolution algorithm. A multi-dimensional indexing strategy for handling large amount of intermediate solutions are proposed as an optimization, which yields considerable efficiency, especially on cases with more than two objectives. A series of computational experiments show the applicability of the proposal to several types of instances.

Keywords: Multi-objective Knapsack Problem, Metaheuristic, Shuffled Complex Evolution, Multi-dimensional indexing

Contents

1 Introdução

Intro...

2 O Problema da Mochila Multi-objetivo

Na vida real é comum a existência de problemas de otimização que consideram mais de um objetivo os quais, geralmente, são conflitantes. Estes problemas são chamados multiobjetivos e tipicamente não possuem solução ótima, ou seja, uma solução que é a melhor em todos os objetivos, mas as soluções de interesse as chamadas *soluções eficientes*.

Um dos problemas multiobjetivos mais importantes é o problema da mochila multi-objetivo (MOKP). Muitas casos reais como seleção de projetos (**??**), orçamento de capital (**??**), carregamento de carga (**??**) and planejamento de estoque (**??**) podem ser modelados como MOKP. O MOKPé considerado um problema $\mathcal{NP}$-Hard, uma vez que é a generalização do problema da mochila $0-1$.

A literatura contém várias propostas para resolver o MOKPde forma exata. Porém, nenhum método tem provado ser eficiente para grande instâncias com mais de dois objetivos. Mesmo para problemas bi-objetivo, algumas instâncias de tamanho considerado médio têm aprestando difculdades na determinação da solução exata, o que tem motivado o desenvolvimento de métodos heurísticas que buscam determinar um conjunto Paretoaproximado em tempo computacional razoável.

O problema da mochila multi-objetivo pode ser descrito como uma função vetorial f que mapeia uma variável de decisão (solução) a uma tupla de m valores (objetivos). Formalmente:

$$\begin{aligned} \max\ \boldsymbol{y} = f(\boldsymbol{x}) &= \big(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\big) \\ \text{sujeito a } \boldsymbol{x} &\in X \end{aligned}$$

onde $\boldsymbol{x}$ é a *variável de decisão*, X denota o conjunto de soluções viáveis e $\boldsymbol{y}$ representa o *vetor de objetivos* para os quais deseja-se maximizar.

Considerando duas soluções $\boldsymbol{a}, \boldsymbol{b} \in X$, $\boldsymbol{a}$ é dita *dominar* $\boldsymbol{b}$ se, e somente se, $\boldsymbol{a}$ é ao menos tão boa quanto $\boldsymbol{b}$ em todos os objetivos e melhor que $\boldsymbol{b}$ em ao menos um dos objetivos. Por motivos práticos diremos que uma solução $\boldsymbol{a}$ domina uma solução $\boldsymbol{b}$ pela notação $dom(\boldsymbol{a}, \boldsymbol{b})$. Formalmente:

$$dom(\boldsymbol{a}, \boldsymbol{b}) = \begin{cases} \forall i \in \{1, 2, \ldots, m\} : f_i(\boldsymbol{a}) \geqslant f_i(\boldsymbol{b}) \text{ and} \\ \exists j \in \{1, 2, \ldots, m\} : f_j(\boldsymbol{a}) > f_j(\boldsymbol{b}) \end{cases}$$

Uma solução viável $\boldsymbol{a} \in X$ é considerada *eficiente* se não é dominada por nenhuma outra solução viável. O conjunto de todas as soluções eficientes de um problema multi-objetivo é chamado de *conjunto Pareto*. Resolver um problema multi-objetivo consiste em determinar seu conjunto Pareto. Vale resaltar que o tamanho do conjunto Pareto-para o problema em questão tende a crescer rapidamente com o tamanho do problema, especialmente com o número de objetivos.

Uma instância de um problema da mochila multi-objetivo (MOKP) com m objetivos consiste em uma capacidade inteira $W > 0$ e n itens. Cada item i possui um peso inteiro positivo w_i e m lucros inteiros $p_i^1, p_i^2, \ldots, p_i^m$ não negativos. O lucro p_i^k representa a contribuição do i-ésimo item para com o k-ésimo objetivo. Uma solução é representada por um conjunto $\boldsymbol{x} \subseteq \{1, \ldots, n\}$ contendo os índices dos itens incluídos na mochila. Uma solução é viável se o peso total incluído na mochila não ultrapassa a capacidade da mochila. Formalmente a definição do problema é a seguinte:

$$\begin{aligned} \max\ f(\boldsymbol{x}) &= \big(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\big) \\ \text{subject to } w(\boldsymbol{x}) &\leqslant W \\ \boldsymbol{x} &\subseteq \{1, \ldots, n\} \end{aligned}$$

where

$$\begin{aligned} f_j(\boldsymbol{x}) &= \sum_{i \in \boldsymbol{x}} p_i^j \quad j = 1, \ldots, m \\ w(\boldsymbol{x}) &= \sum_{i \in \boldsymbol{x}} w_i \end{aligned}$$

O MOKPé considerado um problema $\mathcal{NP}$-Hard visto set uma generalização do bem conhecido problema da mochila $0-1$, para o qual $m = 1$. É consideravelmente difícil determinar o conjunto Paretopara um MOKP, especialmente para vários objetivos. Até mesmo para casos bi-objetivos, problemas pequenos podem se apresentar intratáveis. Por este motivo interessa-se no desenvolvimento de métodos eficientes para manipular uma grande quantidade de soluções, o que pode eventualmente trazer tratabilidade a instâncias antes intratáveis.

Considerando duas soluções $\boldsymbol{x}, \boldsymbol{y} \subseteq \{1, \ldots, n\}$, $\boldsymbol{y}$ é considerada *extensão* de $\boldsymbol{x}$, denotada como $ext(\boldsymbol{y}, \boldsymbol{x})$, se, e somente se, $solx \subseteq \boldsymbol{y}$. Qualquer conjunto $\boldsymbol{e} \subseteq \{1, \ldots, n\}$ tal que $\boldsymbol{x} \cap \boldsymbol{e} = \emptyset$ é dito ser um *extensor* de $\boldsymbol{x}$. Se $w(\boldsymbol{x}) + w(\boldsymbol{e}) \leqslant W$ então $\boldsymbol{e}$ é considerado um *extensor viável* de $\boldsymbol{x}$. Uma solução $\boldsymbol{x}$ é chamada *deficiente* se esta ainda possui capacidade suficiente para incluir um ou mais itens, ou seja, $w(\boldsymbol{x}) + min\{w_i : i \notin \boldsymbol{x}\} \leqslant W$. Considera-se que $\boldsymbol{x}$ *domina segundo a mochila* $\boldsymbol{y}$, denotado por $dom_k(\boldsymbol{x}, \boldsymbol{y})$, se $\boldsymbol{x}$ domina $\boldsymbol{y}$ e não pesa mais que $\boldsymbol{y}$. Formalmente:

$$dom_k(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} dom(\boldsymbol{x}, \boldsymbol{y}) & \text{and} \\ w(\boldsymbol{x}) \leqslant w(\boldsymbol{y}) \end{cases} \tag{2.1}$$

O conceito de dominância segundo a mochila foi primeiramente proposto por Weingartner e Ness (**??**).

A Figura **??** ilustra o conceito para um problema com $m = 1$. Qualquer solução na área hachurada domina segundo a mochila a solução assinalada. Este conceito é largamente utilizado em diversos algoritmo, inclusive no algoritmo exato abordado neste Capítulo.

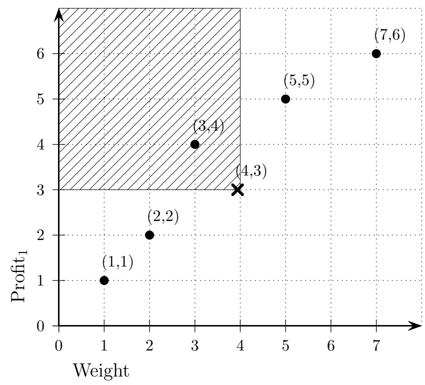

Figura 1 – A knapsack-dominated solution.

2.1 O Problema da Mochila Multi-objetivo Uni-dimensional

O algoritmo de Nemhauser e Ullmann é um algorimto de programação dinâmica que resolve problemas da mochila de forma genérica aplicando o conceito de dominância da mochila para remover soluções parciais que não resultarão em soluções eficientes, ou seja, soluções que irão compor o conjunto Pareto(conjunto solução).

Algorithm 1 O algoritmo de Nemhauser e Ullmann para o MOKP.

function DP$(\boldsymbol{p}, \boldsymbol{w}, W)$
    $S^0 = \left\{\emptyset\right\}$
    for $k \leftarrow 1, n$ **do**
        $S^k_* = S^{k-1} \cup \{\boldsymbol{x} \cup k \mid \boldsymbol{x} \in S^{k-1}\}$ ▷ extensão de soluções
        $S^k = \{\boldsymbol{x} \mid \nexists \boldsymbol{a} \in S^k_* : dom_k(\boldsymbol{a}, \boldsymbol{x})\}$ ▷ filtro de dominância parcial
    end for
    $P = \{\boldsymbol{x} \mid \nexists \boldsymbol{a} \in S^n : dom(\boldsymbol{a}, \boldsymbol{x}) \mid w(\boldsymbol{x}) \leqslant W\}$ ▷ dominância/viabilidade
    return P
end function

3 A k-d tree

4 Experimentos

5 Conclusão