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 problema de otimização multi-objetivo com m objetivos pode ser descrito como uma função vetorial $f(x) = \big(f_1(x), \ldots, f_p(x)\big)$ para a qual deseja-se encontrar um vetor $x \in X$ que maximize simultaneamente as m funções objetivo. Formalmente:

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

Considere um problema de otimização multi-objetivo. Diz-se que uma solução $x \in X$ *domina* uma solução $y \in X$, denotado por $\mathrm{dom}(x, y)$ se, e somente se, x é ao menos tão boa quanto y em todos os objetivos e melhor que y em ao menos um dos objetivos. Formalmente:

$$\mathrm{dom}(x, y) = \begin{cases} \forall i \in \{1, 2, \ldots, m\} : f_i(x) \geqslant f_i(y) \text{ e} \\ \exists j \in \{1, 2, \ldots, m\} : f_j(x) > f_j(y) \end{cases}$$

Uma solução $x \in X$ é dita *eficiente*, denotado por $\mathrm{eff}(x)$, se, e somente se, x não é dominada por nenhuma outra solução pertencente a X. O conjunto de todas as soluções eficientes de um problema multi-objetivo, denotado por $Par(X)$, é chamado de *conjunto Pareto* ou *conjunto Pareto-ótimo*. Formalmente:

$$Par(X) = \{x \in X \mid \mathrm{eff}(x)\}$$

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. Este conceito foi primeiramente elaborado por Vilfredo Pareto em 1896, que enunciou a relação Pareto-Ótima que diz: “não é possível melhorar uma característica do problema sem piorar outra”, o que caracteriza a relação conflitante entre os objetivos na otimização multi-objetivo.

Um dos problemas multiobjetivos mais importantes da literatura é o problema da mochila multiobjetivo (MOKP). Muitas problemas reais podem ser modelados como uma instância do MOKP como seleção de projetos (TENG; TZENG, 1996), orçamento de capital (ROSENBLATT; SINUANY-STERN, 1989), carregamento de carga (TENG; TZENG, 1996) e planejamento de estoque (ISHIBUCHI; AKEDO; NOJIMA, 2015).

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.

Vale resaltar que o tamanho do conjunto Paretopara 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 MOKP é 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

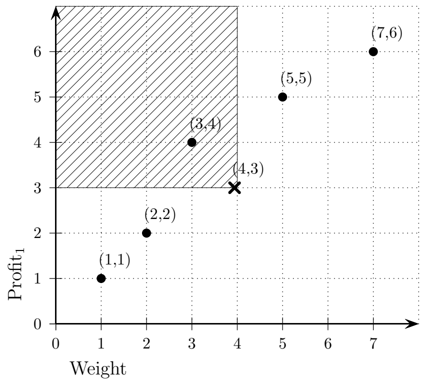

Figura 1 – A knapsack-dominated solution.

$\boldsymbol{y}$ 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 (WEINGARTNER; NESS, 1967).

A Figura 1 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.

A literatura contém várias propostas para resolver o MOKP de 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.

2.1 Métodos Exatos

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).

O algoritmo inicia definindo uma solução inicial S^0 contendo apenas a solução vazia (linha 2). Na k-ésimo iteração o algoritmo recebe um conjunto S^{k-1} contendo soluções exclusivamente compostas pelos primeiros $k-1$ itens, ou seja, $\forall \boldsymbol{x} \in S^{k-1}, \boldsymbol{x} \subseteq \{1, \ldots, k-1\}$. O conjunto S^{k-1} é então expandido adicionando-se uma cópia de cada uma das suas soluções mas desta vez incluindo o k-ésimo item (linha 4), formando o conjunto S_*^k, o qual possui o dobro da cardinalidade de S^{k-1}. O conjunto S_*^k..

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

Apesar de sua simplicidade o Algoritmo 1 é consideravelmente poderoso. Contudo o potencial crescimento exponencial do conjunto Paretopara o MOKPcompromete severamente o seu desempenho. Uma forma de atacar este problema é tentar reduzir ainda mais a quantidade de soluções parciais manuseadas durante as iterações do algoritmo. Três propostas...

2.2 Métodos Heurísticos

3 A k-d tree

4 Experimentos

5 Conclusão

References

ISHIBUCHI, H.; AKEDO, N.; NOJIMA, Y. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 19, n. 2, p. 264–283, 2015.

ROSENBLATT, M. J.; SINUANY-STERN, Z. Generating the discrete efficient frontier to the capital budgeting problem. *Operations Research*, INFORMS, v. 37, n. 3, p. 384–394, 1989.

TENG, J.-Y.; TZENG, G.-H. A multiobjective programming approach for selecting non-independent transportation investment alternatives. *Transportation Research Part B: Methodological*, Elsevier, v. 30, n. 4, p. 291–307, 1996.

WEINGARTNER, H. M.; NESS, D. N. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, INFORMS, v. 15, n. 1, p. 83–103, 1967.