Discrete Optimization

Global optimization and multi knapsack: A percolation algorithm

Dominique Fortin *, Ider Tsevendorj

Inria, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France

Received 12 September 2000; accepted 4 July 2002

Abstract

Since the standard multi knapsack problem, may be rewritten as a reverse convex problem, we present a global optimization approach. It is known from solving high dimensional nonconvex problems that pure cutting plane methods may fail and branch-and-bound is impractical, due to a large duality gap. On the other hand, a strategy based on some sufficient optimality condition does not help much because it requires generating all level set points, an intractable problem. Therefore, we propose to combine both a cutting plane method and a sufficient optimality condition together with a random generation of level set points where the number of points is limited by a tabu list to prevent re-examination of the same level set area. Experiments show that we end up with a small duality gap allowing a subsequent branch-and-bound approach to prove optimality.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Global optimization; Integer programming; Combinatorial optimization; Cutting

1. Introduction

Given an $m \times n$ matrix $A$ with real nonnegative entries, a nonnegative weight vector $c = (c_1, \ldots, c_n)$ and a nonnegative supply vector $b = (b_1, \ldots, b_m)$, the classical multi knapsack problem (MKP) is given by:

$$\begin{align*}
\max & \quad \langle c, x \rangle \\
\text{s.t.} & \quad Ax \leq b, \\
& \quad x \in \{0, 1\}^n,
\end{align*}$$

where each row $a' \in \mathbb{R}^n$ of $A$ with corresponding $b_i$ is a so-called knapsack constraint and we use angles for standard innerproduct. This problem has long been studied, see for instance Osorio et al. [11] for a thorough review.

It is customary to assume each unit vector $e_i$ of canonical basis of $\mathbb{R}^n$ being feasible for (MKP) in order to deal with full dimensional polytope only ($\dim(MKP) = n$). Standard linear relaxation (LRMKP) amounts to turn binary constraints on $x$ into box constraints on $x$:

$$\begin{align*}
\max & \quad \langle c, x \rangle \\
\text{s.t.} & \quad Ax \leq b, \\
& \quad x \in [0, 1]^n,
\end{align*}$$

However, it introduces fractional points useless for original problem so that we keep equivalence only
if we further require feasible points to lay outside an hypersphere close enough to the unit hypersphere (RCMKP) centered at $e/2$ where $e$ stands for the all ones vector.

$$\max \langle e, x \rangle$$

s.t. $Ax \leq b$

$x \in \{0, 1\}^n$,

$g_{\delta}(x) \geq 0$,

where $g_{\delta}(x) = (x - (e/2), x - (e/2)) - (n/4) + \delta$

and we will name it geodesic preferably to level set in order to stress the connection with $\delta$ parameter while level set is tightly connected to an objective value.

Clearly, for $\delta = 0$, both problems are equivalent $(\text{MKP}) \equiv_0 (\text{RCMKP})$; yet, it is well known [14] that for single knapsack problem (SKP), exists $\delta_0$ such that for all $\delta \leq \delta_0$, $(\text{SKP}) \equiv_{\delta} (\text{RCSKP})$. It carries over $(\text{MKP})$ by taking minimum $\delta_0$ over all knapsack constraints; therefore, for $\delta$ small enough: $(\text{MKP}) \equiv_{\delta} (\text{RCMKP})$. The latter form is known as reverse convex optimization.

The main purpose of this paper is to compare a global optimization approach to (MKP) with heuristics performance with respect to speed and quality of solution retrieved. It is known from solving high dimensional nonconvex problems that pure cutting plane methods may fail [6] and branch-and-bound (B&B) is impractical, due to a large duality gap. On the other hand, $\mathfrak{R}$-strategy [13] does not help much because we do not know how to generate all geodesic points. Therefore we propose to combine both a cutting plane method and $\mathfrak{R}$-strategy with a random generation of geodesic points where the number of points is limited by a tabu list to prevent re-examination of the same geodesic area. Experiments show that we end up with a small duality gap allowing a subsequent B&B approach to prove optimality for reasonable sized instances.

In Sections 2 and 3 we use known results from global optimization [10] and tailor them for (MKP); in Section 4, we introduce some equivalence relation to reduce optimality condition checking.

Finally, in Section 5, we give computational results to show how favorably this algorithm compares with standard heuristics [1–4,7,9] or other cutting plane algorithms [11].

2. Optimality conditions

Let a reverse convex problem (RCP) be given by

$$\min f(x)$$

s.t. $x \in \mathcal{F} \subset \mathbb{R}^n$, $g(x) \geq 0$,

under $f, g, \mathcal{F}$ being convex. Constraint $g(x) \geq 0$ is said reverse convex. In the sequel, we assume $g$ to be differentiable for sake of simplicity but it carries over general case by replacing gradient with sub-differential $\partial g$.

Definition 1 (actual reverse convex problem). (RCP) is an actual reverse convex problem if there exists $v \in \mathbb{R}^n$ with $g(v) < 0$; otherwise reverse convex constraint is trivially fulfilled.

Definition 2 (reverse convex problem regularity). (RCP) is regular whenever for all $y \in \mathcal{F}$ such that $g(y) = 0$, there exists $x \in \mathcal{F}$: $\langle \nabla g(y), x - y \rangle > 0$.

Theorem 3 (Strekalovskii [12]). For (RCP) being an actual and regular reverse convex problem, let $z$ be a feasible point

if for all $y : g(y) = 0$ and for all $x \in \mathcal{F}$:

$$f(x) \leq f(z), \langle \nabla g(y), x - y \rangle \leq 0$$

holds then $z$ is a global solution to (RCP).

In order to check the sufficient optimality condition, one has to solve a collection of linearized problems.

for all $y : g(y) = 0$,

$$\max \langle \nabla g(y), x \rangle$$

s.t. $x \in \mathcal{F}$, $f(x) \leq f(z)$

and then check whether

$$\langle \nabla g(y), u(y) - y \rangle \leq 0$$

is satisfied for all $y : g(y) = 0$ at $u(y)$ solution of linearized problem (1).
Claim 4. Let $F = [0, 1]^n \cap Ax \leq b$ be intersection of box constraints and knapsack constraints, then (RCMKP) is both an actual and regular reverse convex problem. Therefore the condition of Theorem 3 is sufficient for global optimality in (RCMKP) under $\delta$ small enough.

**Proof.** $\arg \min g_\delta(x) = e/2$ guarantees first condition ($g_\delta(e/2) = \delta - (n/4) < 0$) while $\delta > 0$ implies existence of required points in curved corners, i.e. points within box and outside hypersphere. □

In the case of (RCMKP) we have sufficient optimality condition of being $z$ global solution:

- for all $y : g_\delta(y) = 0$,
- $\langle \nabla g_\delta(y), x - y \rangle \leq 0$,
- for all $x \in [0, 1]^n$,
- $Ax \leq b$, $-\langle c, x \rangle \leq -\langle c, z \rangle$.

Since it is intractable to check inequality (2) for all $y : g_\delta(y) = 0$, it suggests to sample geodesic by shooting in random directions as long as condition on geodesic point (met along direction) is satisfied; otherwise, we found a better point and continues from this percolated point. Stopping criterion involves maximum number of shots or small gap between upper bound and $(c, z)$ in (RCMKP). We refine this random shooting by first looking at easy (analytically found) geodesic points along box and knapsack constraints and then at random direction as explained later in Section 5.

3. Cutting plane

From global optimization [8], we adapt standard retrieval of a cutting plane to (MKP). Let us assume that $u \in \mathbb{R}^n$ solves problem (LRMKP). Obviously, if $u \in \{0, 1\}^n$ then it solves (MKP) as well, else we have to cut this fractional point.

Let $P = \{x \mid Ax \leq b, x \in [0, 1]^n\}$ and $u = \arg \max_{x \in P} \langle c, x \rangle$ s.t. $g_\delta(u) < 0$ be given. Let $A(u)$ denote the matrix of active constraints at $u$. Then from $A^{-1}(u)$ (inverse of $A(u)$ and Kronecker product $\otimes$, one gets

$Z = u \otimes e^\top + A^{-1}(u)z$

and

$Y = u \otimes e^\top + A^{-1}(u)\beta$

for some $a \in \mathbb{R}^n$ such that every column vector $z_i$ of $Z$ belongs to $\tilde{P}$ and some $\beta \in \mathbb{R}^n$ such that every column vector $y_i$, of $Y$ fulfills $g_\delta(y_i) = 0$.

In Fig. 1, we show how global optimization cut (standard in so-called global approach) could be strengthen by replacing the hypersphere $\delta$ by the best known profit hypersphere known so far (denoted as percolation cut to refer to our geodesic partitioning scheme). Most texts on (MKP) name above profit constraint as the lower bound constraint to explicitly refer to maximization. We discard from our study surrogate constraints as well as constraints pairing to strictly focus on global approach; but efficiency of those techniques are proven for instance in [11] where profit con-
straint is paired with surrogate constraints to yield
nested logic cuts. Fig. 1 right opens up the ques-
tion of relationship among all above cuts either
global or logic.

Claim 5. Let \( Y \) be defined as above and \( d \) be unique
solution of system of linear equations
\[
Yd = ne
\]
then hyperplane \( \langle d, x \rangle = n \) (see Fig. 1 percolation
cut) cuts only fractional points together with \( u \).

Proof. Due to convexity of function \( g_d(\cdot) \) and
\( g_d(y_i) = 0 \) \((i = 1, 2, \ldots, n)\), \( g_d(u) < 0 \)
we have \( g_d(x) \leq 0 \) for all \( x \) such that
\[
x = u \left( 1 - \sum_{i=1}^{n} y_i \right) + \sum_{i=1}^{n} y_i v_i,
\]
\[
\sum_{i=1}^{n} y_i \leq 1, \quad y_i \geq 0 \quad (i = 1, 2, \ldots, n).
\]
It is worthwhile to notice that any constant in
right hand side is convenient; however \( n \) happens
to be numerically stable in our experiments
whereas 1 introduces highly unstable behavior.

4. Geodesic partitioning

In this section, we specialize inverse LP prob-
lems to partition geodesic into equivalence classes
w.r.t. percolation phase. For any LP \( \text{max}_{x} \langle c, x \rangle \) on
set \( P = \{ x \mid Ax \leq b \} \), and feasible solution \( \hat{x} \in P \),
we associate a so-called inverse LP: find \( \hat{c}, \hat{b} \) such
that \( \hat{x} = \text{arg max}_{x} \langle \hat{c}, x \rangle \) on set \( P = \{ x \mid Ax \leq \hat{b} \} \).

Lemma 6. Let \( u \) be a vertex of polytope
\( P = \{ x \mid Ax \leq b \} \) and \( A(u) \) be the set of active con-
straints at \( u \); then for all \( x \geq 0, \langle x, e \rangle = 1, \)
\( u \in \text{arg max}\{ \langle A(u)x, x \rangle \mid Ax \leq b \} \) holds.

Proof. Noting \( d = A(u)x \), we have for all \( x \in P \langle d, x \rangle \leq \langle d, u \rangle \) since \( Au \leq b \) and \( x \geq 0 \).

Remark 7. Percolation or global optimization cuts
in Section 3 come from inverse LP where the ac-
tual depth of hyperplanes is related to corre-
sponding geodesics.

Corollary 8. For a full dimensional polytope
\( P = \{ x \mid Ax \leq b, x \in [0, 1]^n \} \) and reverse convex
problem \( \text{min}_{x} f(x) \text{ s.t. } x \in P, g(x) \geq 0 \), let \( u \) be a
solution of linear relaxation \( \text{min}_{x \in \mathbb{R}^n} f(x) \) on re-
laxed polytope \( P_{LR} = \{ x \mid Ax \leq b \} \), then \( u \) is repre-
sentative for all \( x \in P \) which is a strictly convex
combination of active constraints at \( u \), i.e. \( x = A(u)x \)
for \( x \) strictly inside the box \([0, 1]^n\).

Proof. Since polytope is full dimensional,
\( \text{arg min}\{ \langle A(u)x, x \rangle \mid Ax \leq b \} \) is unique. \( \square \)

Since (MKP) is full dimensional, it suggests to
introduce some tabu list along random generation
of geodesic points; let \( y, g_d(y) = 0 \) a geodesic point
generated by random direction \( d \), then \( d \) is tabu if
it is convex combination of active direction \( A(u) \)
for some \( u \) solution of linear relaxation (LRMKP).
To be more precise,
\[
y = \frac{d}{2} + \frac{n - 4\delta}{\|d\|^2} \cdot d
\]
for convex combination \( d \) at \( u \) leads to already
examined geodesic area.

5. Computational results

In this section, we give a more detailed de-
scription of the algorithm and report the behavior
of the main feature of reverse convex approach
along with two different enhancements: intensifi-
cation technique compared with pure branch-and-
cut technique on celebrated samples.

5.1. A percolation algorithm

For sake of conciseness, we use the outerprod-
uct \( u \wedge v \) notation for colinearity testing and \( [u] \)
(resp. \( \lfloor u \rfloor \) ) for rounding to the nearest integer to \( u \)
(resp. nearest integer below \( u \) ) componentwise; we
use lp_solve as a generic linear solver but all
experiments were done under CPLEX \footnote{1} and

\footnote{1} CPLEX is a registered trademark of ILOG Copyright ©
ABACUS\textsuperscript{2} to manage constraint and cutting planes. In the pseudo code MKP returns the best solution found, SEPARATE returns a set of cutting planes at \( u \) if any and PERCOLATE returns a better point found at \( u \) or \( u \) itself.

\[
\text{MKP}(\max \langle c, x \rangle, Ax \leq b, x \in \{0, 1\}^n)
\]

1. \textbf{repeat}
2. \( u := \text{lp.solver}(\max \langle c, x \rangle, Ax \leq b, x \in \{0, 1\}^n))\)
3. \textbf{if} \( z = [u] \) or \( z = [u] \) \textbf{better solution}
4. \textbf{then} add \(-\langle c, x \rangle \leq -\langle c, z \rangle, x \in \{0, 1\}^n\));
5. \textbf{end if}
6. \textbf{trivial cutting *}
7. \textbf{unti} SEPARATE\( (u, A(u)) \neq \emptyset;\) cutting plane *
8. \textbf{return} \( z; \)

where \( A(u) \) stands for active constraints at \( u \).

\[
\text{SEPARATE}(u, A(u))
\]

1. \( \text{URAY} := A(u)____^{-1};\) extreme rays of active cone *
2. \( U := \emptyset;\) geodesic points from active cone *
3. \( V := \emptyset;\) feasible adjacent vertices along extreme rays *
4. \textbf{foreach ray in URAY do}
5. \( \text{find } y \text{ s.t. } \{ (y - U) \land \text{ray} = 0, g_\delta(y) = 0 \}
6. \( \text{if } (z = [y] \text{ or } z = [y]) \text{ better solution}
7. \( \text{then}
8. \( u := \text{lp.solver}(\max \langle c, x \rangle, Ax \leq b, -\langle c, x \rangle \leq -\langle c, z \rangle, x \in \{0, 1\}^n))\); \textbf{end if}
9. \( \text{trivial cutting *}
10. \textbf{return} \text{SEPARATE}(u, A(u));\) cutting plane *
11. \textbf{else}
12. \( U := U \cup y; \)
13. \( V := V \cup z;\text{s.t. } \{(z - u) \land (y - u) = 0, z \in (LRMKP)\}
14. \textbf{endif}
15. \textbf{end for}
16. \textbf{U is cutting plane by above *}
17. \( l := \text{arg min}\langle c, v \rangle \text{ over } v \in V;\) lower bound simplex *
18. \( \text{LRAY} := A(l)____^{-1};\) extreme rays at lower bound cone *
19. \( L := \emptyset;\) geodesic points from lower bound cone *

\[
\text{PERCOLATE}(y, u)
\]

1. \( z := u(\delta)/\) geodesic point close to \( u \) corner *
2. \( \tilde{x} := \{ \max (\nabla g_\delta(y), x), Ax \leq b, -\langle c, x \rangle \leq -\langle c, z \rangle, x \in \{0, 1\}^n)\}
3. \textbf{foreach } \langle e, \tilde{x} \rangle = 0 \text{ or } \langle e, \tilde{x} \rangle = 1 \text{ do}
4. \( v := \text{sibling box constraint point } l^* \text{ n such points *}
5. \( \textbf{if } (v) \text{ better point then return } v \text{ endif;}
6. \textbf{end for}
7. \textbf{foreach } A(u) \cup \{-\langle c, x \rangle \leq -\langle c, z \rangle\} \text{ active constraints do}
8. \( \text{find } v \text{ s.t. } \{||u - v|| \text{ maximum, } g_\delta(v) = 0 \}
9. \( \textbf{if } (v) \text{ better point then return } v(\delta) \text{ endif;}
10. \textbf{end for}
11. \textbf{for } l = 1 \text{ to max iteration do}
12. \( g := \text{generate random direction } d \text{ not in tabu list;}
13. \( v := \text{farthest feasible point from } z \text{ along } d; \)
14. \( \text{find } v_1, v_2 \text{ s.t. } \{g_\delta(v_1) = 0, g_\delta(v_2) = 0, (v_2 - v_1) \land (z, v) = 0\}
15. \( \textbf{if } (v_i, i = 1, 2) \text{ is a better point then return } v_i(\delta) \text{ endif;}
16. \textbf{end for}
17. \textbf{return } u; \) no better point found *

\textsuperscript{2} ABACUS a software framework for the development of optimization algorithms distributed by OREAS \url{http://www.oreas.de/frames.html}.
where \( u(\delta), \) \( v(\delta) \) mean that a feasible geodesic point is retrieved from lower bound \( l \) corner along max profit component (in order to increase current lower bound \( (c, l) \) by a small amount depending on \( \delta \)) and where random direction is projected onto active constraint at \( v \) namely \( d_{l,(a,x),(a,x)} \) before retrieving \( v1, v2 \) geodesic farthest points from \( v. \)

5.2. Remarks

Previous algorithm belongs to the class of cutting plane algorithms since it alternates cutting phase with percolation phase; however percolation phase should not be compared with primal improvement heuristics since it relies on global optimality sufficient condition instead of local search improvement.

Let notice that better point in percolate algorithm is related to problem linearized from \((RCMKP)\) instead of profit function in separation phase of outermost algorithm loop \((LRMKP)\) while constraint remain the same.

In SEPARATE algorithm, profit constraint seems redundant in linear programs (lines 8 and 22) but it helps as a stopping criterion when duality gap is small (this early stop was forecasted as a generic feature of global optimization, compared to linear relaxation, in Section 2).

In line 4 of SEPARATE algorithm, we select only one vertex from simplex defined by apex \( u \) the current best fractional solution to \((LRMKP)\) and \( n \) feasible adjacent vertices to \( u. \) In our first attempts, we included all simplicial vertices cuts at the cost of \( n + 1 \) \((n \times n)\) matrices to invert; then we restricted to upper and lower cuts only and save one order of magnitude in computation time without losing accuracy in results. A possible interpretation for no loss of accuracy lay on the compromise between better approximation of polytope \((n + 1 \text{ cuts})\) and less numerous explicit fractional points introduced by 2 cuts.

Another consequence of previous remark on tradeoff between accuracy of polytope approximation and number of explicit fractional points is that we observed better solutions whenever we selected only knapsack cuts (normal in positive orthant) instead of deep cuts with possibly negative direction. However, this effect is blurred by the fact that shooting in knapsack supporting hyperplane is likely to give better geodesic points.

Since it is intractable to check every area associated with a solution, we borrow techniques from tabu search [5] to look for a good direction to search for or to escape from an area we could not improve after some iteration steps; the name percolation algorithm comes after this reduction of the set of areas actually checked.

In line 12 of PERCOLATE algorithm, we used a tabu list (see Section 4) remembering last \( n \) fractional points as a vector of booleans tagging which constraints are active in this point; thus, it requires no more than \( n/32 \) words on most computers for each entry. We did not further study tabu list size on algorithm behavior. Algorithm’s naming comes from percolation of profit constraint through equivalence classes on geodesic i.e. sieving along with objective improvement.

We also refine random direction generation by reminding an average random direction and use a convex combination \((1 - \alpha)\) times average direction plus \( \alpha \) times actual random direction generated; starting with \( \alpha = 1 \) (pure diversification), each maxiteration loops, \( \alpha \) is halved up to 0.05 so that intensification becomes more and more effective.

5.3. Intensification/diversification results

Our experiments are done on mknapcbl.txt, a collection of test data sets from http://mscmga.ms.ic.ac.uk/jeb/orlib/mknapinfo.html.

Maximum number of iteration loops before increasing intensification is set to 100. We report the following indicators in Table 1:

- first sol. = value obtained after first cutting round; in other words, no percolate step has been ever applied until this state,
- final sol. = value at stopping criterion,
- first gap = (firstsol. – bestknown)/max(first sol., best known) in percentage,
- final gap = (finalsol. – bestknown)/max(final sol., best known) in percentage,
- dual, gap = (finalsol. – upperbound)/upper bound in percentage,
- time is from DEC PWS500 workstation,
Table 1

<table>
<thead>
<tr>
<th>#Pb.</th>
<th>First sol.</th>
<th>Final sol.</th>
<th>Best known</th>
<th>First gap (%)</th>
<th>Final gap (%)</th>
<th>Upper bound</th>
<th>Dual. gap</th>
<th>Time (second)</th>
<th>#Percolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>23,061</td>
<td>23,728</td>
<td>24,381</td>
<td>-5.41</td>
<td>-2.67</td>
<td>24,584.3</td>
<td>-3.48</td>
<td>1955</td>
<td>1200</td>
</tr>
<tr>
<td>1</td>
<td>23,818</td>
<td>23,818</td>
<td>24,274</td>
<td>-1.87</td>
<td>-1.87</td>
<td>24,536</td>
<td>-2.92</td>
<td>13,697</td>
<td>1200</td>
</tr>
<tr>
<td>2</td>
<td>22,102</td>
<td>23,257</td>
<td>23,551</td>
<td>-6.15</td>
<td>-1.24</td>
<td>23,893.3</td>
<td>-2.66</td>
<td>1960</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>20,943</td>
<td>22,404</td>
<td>23,534</td>
<td>-11</td>
<td>-4.8</td>
<td>23,717.7</td>
<td>-5.53</td>
<td>5941</td>
<td>1200</td>
</tr>
<tr>
<td>5</td>
<td>24,339</td>
<td>24,339</td>
<td>24,613</td>
<td>-1.11</td>
<td>-1.11</td>
<td>24,872.9</td>
<td>-2.14</td>
<td>4904</td>
<td>1200</td>
</tr>
<tr>
<td>6</td>
<td>23,037</td>
<td>25,351</td>
<td>25,591</td>
<td>-9.98</td>
<td>-0.93</td>
<td>25793.4</td>
<td>-1.71</td>
<td>993</td>
<td>1200</td>
</tr>
<tr>
<td>7</td>
<td>21,543</td>
<td>22,855</td>
<td>23,410</td>
<td>-7.97</td>
<td>-2.37</td>
<td>23,651.9</td>
<td>-3.36</td>
<td>3917</td>
<td>1200</td>
</tr>
<tr>
<td>8</td>
<td>21,814</td>
<td>23,127</td>
<td>24,216</td>
<td>-9.91</td>
<td>-4.49</td>
<td>24,445.6</td>
<td>-3.59</td>
<td>990</td>
<td>1200</td>
</tr>
<tr>
<td>9</td>
<td>22,736</td>
<td>23,826</td>
<td>24,411</td>
<td>-6.86</td>
<td>-2.39</td>
<td>24,635.7</td>
<td>-3.28</td>
<td>994</td>
<td>1201</td>
</tr>
<tr>
<td>10</td>
<td>40,758</td>
<td>42,705</td>
<td>42,757</td>
<td>-4.67</td>
<td>-0.12</td>
<td>42,931.9</td>
<td>-0.52</td>
<td>1961</td>
<td>1201</td>
</tr>
<tr>
<td>11</td>
<td>40,070</td>
<td>42,246</td>
<td>42,545</td>
<td>-5.81</td>
<td>-0.7</td>
<td>42,706.7</td>
<td>-1.07</td>
<td>994</td>
<td>1200</td>
</tr>
<tr>
<td>12</td>
<td>40,875</td>
<td>41,755</td>
<td>41,968</td>
<td>-2.6</td>
<td>-0.5</td>
<td>42,164.1</td>
<td>-0.97</td>
<td>1959</td>
<td>1200</td>
</tr>
<tr>
<td>13</td>
<td>43,512</td>
<td>44,680</td>
<td>45,090</td>
<td>-3.49</td>
<td>-0.9</td>
<td>45,341</td>
<td>-1.45</td>
<td>1964</td>
<td>1200</td>
</tr>
<tr>
<td>14</td>
<td>41,229</td>
<td>41,764</td>
<td>42,218</td>
<td>-2.34</td>
<td>-1.07</td>
<td>42,427.1</td>
<td>-1.56</td>
<td>1978</td>
<td>1200</td>
</tr>
<tr>
<td>15</td>
<td>42,271</td>
<td>42,681</td>
<td>42,927</td>
<td>-1.52</td>
<td>-0.57</td>
<td>43,081</td>
<td>-0.92</td>
<td>1965</td>
<td>1200</td>
</tr>
<tr>
<td>16</td>
<td>41,980</td>
<td>41,980</td>
<td>42,009</td>
<td>-0.06</td>
<td>-0.06</td>
<td>42,185.6</td>
<td>-0.48</td>
<td>2933</td>
<td>1200</td>
</tr>
<tr>
<td>17</td>
<td>43,563</td>
<td>44,704</td>
<td>45,020</td>
<td>-3.23</td>
<td>-0.7</td>
<td>45,261.5</td>
<td>-1.23</td>
<td>2933</td>
<td>1200</td>
</tr>
<tr>
<td>18</td>
<td>42,935</td>
<td>43,122</td>
<td>43,441</td>
<td>-1.16</td>
<td>-0.73</td>
<td>43,566.3</td>
<td>-1.01</td>
<td>2938</td>
<td>1201</td>
</tr>
<tr>
<td>19</td>
<td>44,511</td>
<td>44,511</td>
<td>44,554</td>
<td>-0.09</td>
<td>-0.09</td>
<td>44,795.3</td>
<td>-0.63</td>
<td>1973</td>
<td>1200</td>
</tr>
<tr>
<td>20</td>
<td>58,466</td>
<td>59,642</td>
<td>59,822</td>
<td>-2.26</td>
<td>-0.3</td>
<td>60,016.6</td>
<td>-0.62</td>
<td>993</td>
<td>1200</td>
</tr>
<tr>
<td>21</td>
<td>60,520</td>
<td>61,805</td>
<td>62,081</td>
<td>-2.51</td>
<td>-0.44</td>
<td>62,224.6</td>
<td>-0.67</td>
<td>1965</td>
<td>1200</td>
</tr>
<tr>
<td>22</td>
<td>58,363</td>
<td>59,297</td>
<td>59,802</td>
<td>-2.4</td>
<td>-0.84</td>
<td>59,959.6</td>
<td>-1.1</td>
<td>6863</td>
<td>1200</td>
</tr>
<tr>
<td>23</td>
<td>60,151</td>
<td>60,151</td>
<td>60,479</td>
<td>-0.54</td>
<td>-0.54</td>
<td>60,642.4</td>
<td>-0.81</td>
<td>1965</td>
<td>1200</td>
</tr>
<tr>
<td>24</td>
<td>59,747</td>
<td>60,991</td>
<td>61,091</td>
<td>-2.19</td>
<td>-1.16</td>
<td>61,330.1</td>
<td>-0.55</td>
<td>1967</td>
<td>1200</td>
</tr>
<tr>
<td>25</td>
<td>57,915</td>
<td>58,920</td>
<td>58,959</td>
<td>-1.77</td>
<td>-0.06</td>
<td>59,161.9</td>
<td>-0.4</td>
<td>991</td>
<td>1200</td>
</tr>
<tr>
<td>26</td>
<td>61,383</td>
<td>61,383</td>
<td>61,538</td>
<td>-0.25</td>
<td>-0.25</td>
<td>61,692.5</td>
<td>-0.5</td>
<td>3894</td>
<td>1200</td>
</tr>
<tr>
<td>27</td>
<td>61,184</td>
<td>61,366</td>
<td>61,520</td>
<td>-0.54</td>
<td>-0.25</td>
<td>61,718.9</td>
<td>-0.57</td>
<td>2937</td>
<td>1200</td>
</tr>
<tr>
<td>28</td>
<td>57,638</td>
<td>59,298</td>
<td>59,453</td>
<td>-3.05</td>
<td>-0.26</td>
<td>59,610.5</td>
<td>-0.52</td>
<td>10,756</td>
<td>1200</td>
</tr>
<tr>
<td>29</td>
<td>59,650</td>
<td>59,650</td>
<td>59,965</td>
<td>-0.52</td>
<td>-0.52</td>
<td>60,230.5</td>
<td>-0.96</td>
<td>2938</td>
<td>1200</td>
</tr>
</tbody>
</table>

- #percolation = number of geodesic points examined (on box, knapsack or randomly generated).

Comparison between first gap and final gap shows that both cutting and percolation are efficient. While algorithm affords to interleave cutting and percolation, no improvement of upper bound was observed after first cutting round; on the other hand, percolation roughly halves the gap after intensification process.

5.4. Branch and cut results

First gap in previous experiment was big enough to apply intensification and hopefully we reported efficiency of our percolation heuristic; in order to further reduce the duality gap, it suggests to try a branch and cut algorithm instead. In sample tests below, intensification is completely discarded and branching is done on close half expensive variable (closer to 0.5) after each round of percolation. Our experiments are done on mknaps2.txt, a collection of test data sets from http://mscma.ms.ic.ac.uk/jeb/orlib/mknapinfo.html since optimal solution are known.

Maximum number of iteration loops for percolation is now set to \( n \). We keep meaningful indicators from previous test and add the following for Table 2:

- Problem = name and size as \((m \times n)\) constraints \times variables,
- first (sec) = solution after first cutting round and time in seconds on DEC PWS500 workstation,
### Table 2
Branch and cut under percolation on mknaps2.txt

<table>
<thead>
<tr>
<th>Problem</th>
<th>First (second)</th>
<th>Opt. sol.</th>
<th>First gap (%)</th>
<th>Upper bound</th>
<th>Dual. gap (%)</th>
<th>Time (second)</th>
<th>#Level</th>
<th>#Sub</th>
<th>#Percolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENTO1.DAT(30 x 60)</td>
<td>7761 (27)</td>
<td>7772</td>
<td>-0.14</td>
<td>7780.37</td>
<td>-0.1</td>
<td>2373</td>
<td>40</td>
<td>631</td>
<td>75,600</td>
</tr>
<tr>
<td>SENTO2.DAT(30 x 60)</td>
<td>8716 (41)</td>
<td>8722</td>
<td>-0.06</td>
<td>8723.85</td>
<td>-0.02</td>
<td>5828</td>
<td>35</td>
<td>1363</td>
<td>163,440</td>
</tr>
<tr>
<td>WEING1.DAT(2 x 28)</td>
<td>141,278 (1)</td>
<td>141,278</td>
<td>0</td>
<td>141,332</td>
<td>-0.03</td>
<td>33</td>
<td>17</td>
<td>91</td>
<td>5043</td>
</tr>
<tr>
<td>WEING2.DAT(2 x 28)</td>
<td>130,883 (1)</td>
<td>130,883</td>
<td>0</td>
<td>130,929</td>
<td>-0.03</td>
<td>16</td>
<td>17</td>
<td>49</td>
<td>2691</td>
</tr>
<tr>
<td>WEING3.DAT(2 x 28)</td>
<td>95,677 (1)</td>
<td>95,677</td>
<td>0</td>
<td>95,691.2</td>
<td>-0.01</td>
<td>27</td>
<td>24</td>
<td>81</td>
<td>4480</td>
</tr>
<tr>
<td>WBING4.DAT(2 x 28)</td>
<td>115,687 (2)</td>
<td>119,337</td>
<td>-3.05</td>
<td>119,370</td>
<td>-0.02</td>
<td>44</td>
<td>16</td>
<td>119</td>
<td>5936</td>
</tr>
<tr>
<td>WEING5.DAT(2 x 28)</td>
<td>98,016 (1)</td>
<td>98,796</td>
<td>-0.78</td>
<td>98,840</td>
<td>-0.04</td>
<td>13</td>
<td>19</td>
<td>43</td>
<td>1568</td>
</tr>
<tr>
<td>WEING6.DAT(2 x 28)</td>
<td>130,233 (1)</td>
<td>130,623</td>
<td>-0.29</td>
<td>130,634</td>
<td>0</td>
<td>27</td>
<td>12</td>
<td>79</td>
<td>4386</td>
</tr>
<tr>
<td>WEING7.DAT(2 x 105)</td>
<td>1,095,360 (33)</td>
<td>1,095,445</td>
<td>0</td>
<td>1,095,460</td>
<td>0</td>
<td>1002</td>
<td>16</td>
<td>79</td>
<td>16,419</td>
</tr>
<tr>
<td>WEING8.DAT(2 x 105)</td>
<td>609,580 (56)</td>
<td>624,319</td>
<td>-2.36</td>
<td>628,774</td>
<td>-0.7</td>
<td>10,086</td>
<td>77</td>
<td>829</td>
<td>173,880</td>
</tr>
<tr>
<td>WEISH01.DAT(5 x 30)</td>
<td>4554 (4)</td>
<td>4554</td>
<td>0</td>
<td>4561.63</td>
<td>-0.16</td>
<td>82</td>
<td>21</td>
<td>117</td>
<td>6960</td>
</tr>
<tr>
<td>WEISH05.DAT(5 x 30)</td>
<td>4506 (1)</td>
<td>4536</td>
<td>-0.66</td>
<td>4534.4</td>
<td>-0.16</td>
<td>22</td>
<td>10</td>
<td>33</td>
<td>1693</td>
</tr>
<tr>
<td>WEISH06.DAT(5 x 30)</td>
<td>4506 (1)</td>
<td>4115</td>
<td>-0.66</td>
<td>4137.99</td>
<td>-0.55</td>
<td>38</td>
<td>25</td>
<td>79</td>
<td>4680</td>
</tr>
<tr>
<td>WEISH07.DAT(5 x 30)</td>
<td>4561 (1)</td>
<td>4561</td>
<td>0</td>
<td>4570.81</td>
<td>-0.21</td>
<td>21</td>
<td>13</td>
<td>31</td>
<td>1800</td>
</tr>
<tr>
<td>WEISH09.DAT(5 x 30)</td>
<td>4514 (1)</td>
<td>4514</td>
<td>0</td>
<td>4530.78</td>
<td>-0.37</td>
<td>14</td>
<td>15</td>
<td>29</td>
<td>1680</td>
</tr>
<tr>
<td>WEISH10.DAT(5 x 30)</td>
<td>5557 (3)</td>
<td>5557</td>
<td>0</td>
<td>5571.9</td>
<td>-0.26</td>
<td>119</td>
<td>31</td>
<td>111</td>
<td>8804</td>
</tr>
<tr>
<td>WEISH11.DAT(5 x 30)</td>
<td>5567 (6)</td>
<td>5567</td>
<td>0</td>
<td>5567.52</td>
<td>0</td>
<td>210</td>
<td>23</td>
<td>117</td>
<td>9280</td>
</tr>
<tr>
<td>WEISH13.DAT(5 x 40)</td>
<td>5603 (7)</td>
<td>5605</td>
<td>-0.03</td>
<td>5605.79</td>
<td>-0.01</td>
<td>110</td>
<td>21</td>
<td>93</td>
<td>7364</td>
</tr>
<tr>
<td>WEISH14.DAT(5 x 40)</td>
<td>5246 (2)</td>
<td>5246</td>
<td>0</td>
<td>5247.13</td>
<td>-0.02</td>
<td>39</td>
<td>22</td>
<td>45</td>
<td>3520</td>
</tr>
<tr>
<td>WEISH15.DAT(5 x 40)</td>
<td>6339 (7)</td>
<td>6339</td>
<td>0</td>
<td>6370.79</td>
<td>-0.49</td>
<td>493</td>
<td>34</td>
<td>289</td>
<td>28,800</td>
</tr>
<tr>
<td>WEISH16.DAT(5 x 50)</td>
<td>6563 (3)</td>
<td>6563</td>
<td>0</td>
<td>6564</td>
<td>-0.01</td>
<td>178</td>
<td>32</td>
<td>109</td>
<td>10,800</td>
</tr>
<tr>
<td>WEISH17.DAT(5 x 50)</td>
<td>6339 (3)</td>
<td>6339</td>
<td>0</td>
<td>6357.26</td>
<td>-0.28</td>
<td>222</td>
<td>31</td>
<td>133</td>
<td>13,200</td>
</tr>
<tr>
<td>WEISH18.DAT(5 x 50)</td>
<td>6159 (7)</td>
<td>6159</td>
<td>0</td>
<td>6159.65</td>
<td>-0.01</td>
<td>171</td>
<td>35</td>
<td>103</td>
<td>10,200</td>
</tr>
<tr>
<td>WEISH19.DAT(5 x 60)</td>
<td>6954 (6)</td>
<td>6954</td>
<td>0</td>
<td>6955.29</td>
<td>-0.01</td>
<td>298</td>
<td>35</td>
<td>119</td>
<td>14,160</td>
</tr>
<tr>
<td>WEISH20.DAT(5 x 60)</td>
<td>7486 (6)</td>
<td>7486</td>
<td>0</td>
<td>7486.82</td>
<td>-0.01</td>
<td>215</td>
<td>22</td>
<td>91</td>
<td>0</td>
</tr>
<tr>
<td>WEISH21.DAT(5 x 60)</td>
<td>7273 (8)</td>
<td>7289</td>
<td>-0.21</td>
<td>7289.36</td>
<td>0</td>
<td>382</td>
<td>38</td>
<td>153</td>
<td>18,240</td>
</tr>
<tr>
<td>WEISH23.DAT(5 x 60)</td>
<td>8624 (6)</td>
<td>8633</td>
<td>-0.1</td>
<td>8635.83</td>
<td>-0.03</td>
<td>109</td>
<td>13</td>
<td>37</td>
<td>4321</td>
</tr>
<tr>
<td>WEISH24.DAT(5 x 60)</td>
<td>9565 (30)</td>
<td>9580</td>
<td>0</td>
<td>9585.67</td>
<td>-0.05</td>
<td>1528</td>
<td>33</td>
<td>319</td>
<td>44,587</td>
</tr>
<tr>
<td>WEISH25.DAT(5 x 70)</td>
<td>9441 (10)</td>
<td>9450</td>
<td>-0.09</td>
<td>9452.03</td>
<td>-0.02</td>
<td>598</td>
<td>37</td>
<td>139</td>
<td>19,320</td>
</tr>
<tr>
<td>WEISH26.DAT(5 x 70)</td>
<td>9040 (11)</td>
<td>9074</td>
<td>-0.37</td>
<td>9110.5</td>
<td>-0.4</td>
<td>1032</td>
<td>51</td>
<td>247</td>
<td>34,440</td>
</tr>
<tr>
<td>WEISH27.DAT(5 x 80)</td>
<td>8914 (22)</td>
<td>8947</td>
<td>-0.36</td>
<td>9004.18</td>
<td>-0.63</td>
<td>1610</td>
<td>55</td>
<td>261</td>
<td>41,664</td>
</tr>
<tr>
<td>WEISH28.DAT(5 x 80)</td>
<td>8341 (32)</td>
<td>8344</td>
<td>-0.03</td>
<td>8344.58</td>
<td>0</td>
<td>1191</td>
<td>45</td>
<td>193</td>
<td>30,720</td>
</tr>
<tr>
<td>WEISH29.DAT(5 x 80)</td>
<td>10,192 (20)</td>
<td>10,220</td>
<td>-0.27</td>
<td>10,223</td>
<td>-0.02</td>
<td>1222</td>
<td>46</td>
<td>197</td>
<td>31,360</td>
</tr>
<tr>
<td>WEISH30.DAT(5 x 90)</td>
<td>9875 (28)</td>
<td>9939</td>
<td>-0.64</td>
<td>9952.1</td>
<td>-0.13</td>
<td>3552</td>
<td>48</td>
<td>611</td>
<td>39,995</td>
</tr>
<tr>
<td>PB1.DAT(4 x 27)</td>
<td>3034 (3)</td>
<td>3090</td>
<td>-1.81</td>
<td>3100.21</td>
<td>-0.32</td>
<td>15</td>
<td>13</td>
<td>35</td>
<td>972</td>
</tr>
<tr>
<td>PB2.DAT(4 x 34)</td>
<td>2974 (13)</td>
<td>3186</td>
<td>-6.65</td>
<td>3190.27</td>
<td>-0.13</td>
<td>267</td>
<td>16</td>
<td>305</td>
<td>9299</td>
</tr>
</tbody>
</table>
• opt. sol. = optimal solution,
• \#sub = number of subproblems in B&B tree,
• \#level = highest level in B&B tree,
• \#percolation = total number of geodesic points generated.

Apart from PB4.DAT sample whose first gap is rather bad but ending with a gap of 5%, the first solution is slightly improved at a heavy cost in terms of subproblems generated and a moderate number of levels in the B&B tree (30 levels on average with a worse case 77).

6. Concluding remarks

First of all, we addressed (MKP) from a global optimization viewpoint; more precisely we turn it into a reverse convex problem. We design a cutting plane algorithm enhanced with an intensification/diversification technique borrowed from tabu search in order to scan a larger area of possibly good solutions. However, we will stress how further shed could be put on the convexification process at the theoretical level and on the computational tradeoffs to tune parameters (#cutting planes vs. #percolation points).

6.1. Dual Lagrangian vs. global cut

Percolation algorithm proposed in this paper, only relies on inside cut: either global cut (associated with hypersphere \( \delta \)) or percolation cut (associated with best profit hypersphere). However, there is an outside counterpart of inside cut approach; from box constraints \( 0 \leq x \leq 1 \), we could introduce redundant quadratic constraints \( \langle x, x - e \rangle \leq 0 \) into (RCMKP) and then take the Lagrangian to get (LRCMKP).

\[
\min_x \quad L(x, \lambda, \rho) \\
\text{s.t.} \\
Ax \leq b, \\
x \in [0, 1]^n, \\
g_\delta(x) \geq 0, \\
\lambda \in \mathbb{R}_+, \quad \rho \in \mathbb{R}_m^m.
\]
where
\[ \mathcal{L}(x, \lambda, \rho) = -(c, x) + \lambda(x, x - e) + (\rho, Ax - b) = \lambda\|x - x(\lambda, \rho)\|^2 - [\lambda]\|x(\lambda, \rho)\|^2 + (\rho, b) \]

with
\[ x(\lambda, \rho) = \frac{e}{2} + \frac{1}{2\lambda}(c - A^T\rho). \]

The Lebesgue set \( \{x \mid \mathcal{L}(x, \lambda, \rho) \leq \mathcal{L}(z, \lambda, \rho)\} = \{x \mid \|x - x(\lambda, \rho)\| \leq r_x\} \) which contains minimum of \((\text{RCMKP})\) is the ball with radius \( r_x = \|z - x(\lambda, \rho)\| \) and center \( x(\lambda, \rho) \). On the other hand we have reverse convex constraint \( g_8(x) \geq 0 \) which removes interior of the ball with radius \( r_g = ((n/4) - \delta)^2 \) and center \( e/2 \).

Then we define Lagrangian cut as following.

**Claim 9.** For all \( v \in \{x \mid \|x - x(\lambda, \rho)\|^2 \leq r_x^2\} \setminus \{x \mid \|x - \frac{e}{2}\|^2 < r_g^2\} \):
\[ (c - A^T\rho, v) \geq \lambda\left(\frac{1}{2\lambda}\|c - A^T\rho\|^2 - r_x^2 + r_g^2\right). \]

**Remark 10.** Lagrangian cut in the unknowns \( \lambda, \rho \) can be, for example, fully determined from the solution of the following system of equations:
\[ Ax(\lambda, \rho) = b, \]
\[ \langle c, x(\lambda, \rho) \rangle = \langle c, z \rangle. \]

Let \( \gamma = 1/2\lambda, \sigma = -(1/2\lambda)\rho \) replace unknowns \( \lambda, \rho \), then it leads to \((m + 1) \times (m + 1)\) linear system:
\[
\begin{align*}
\gamma c + AA^T\sigma &= b - \frac{1}{2}Ae, \\
\gamma\|c\|^2 + \langle c, A\sigma \rangle &= \left\langle c, z - \frac{e}{2} \right\rangle,
\end{align*}
\]
whose solution is unique from definite positiveness of matrix \( AA^T \) and \( \|c\|^2 > 0 \). In Fig. 2, both cuts are drawn and we may expect that for some \( \lambda, \rho \) lagrangian cut may outperform percolation cut (see Section 3).

We could notice, furthermore, that standard Lagrangian rule in addition to \( \partial \mathcal{L}(x, \lambda, \rho)/\partial x = 0 \) requires \( \partial \mathcal{L}(x, \lambda, \rho)/\partial \rho \geq 0 \) and \( \partial \mathcal{L}(x, \lambda, \rho)/\partial \lambda \geq 0 \); the latter yields \( g_8 \geq 0 \), which points out how Lagrangian relaxation turns a minimization problem into a reverse convex problem (at first sight (RCMKP) derivation appears tricky but its flavor comes from Lagrangian dual on \( \langle x, x - e \rangle \)).

If surrogate constraints belong to inside cuts class, nested logic cuts are rather from the outside cuts class but relationships among both of them was outside our scope and merit a dedicated study.

### 6.2. Computational tradeoffs

For sake of comparison results, we consider as starting feasible solution one obtained by pure cutting plane initial phase; of course, any primal heuristics giving a good starting point in a very small amount of time could be plugged on top of above algorithm to improve running time. We guess that it will improve numerical stability as well, since fewer cutting planes would be introduced whenever profit constraint is good enough.

### 6.3. Further research

Our scope was a case study of global optimization approach to a difficult combinatorial problem; our experiments lead us to strongly argue that this percolation point of view carries over any combinatorial problem in binary variables under the less restrictive assumption of non full dimensionality; it requires dedicated geodesic (or level set) partitioning as well as cutting plane lifting. While lifting sounds harder than partitioning, both seems achievable; then, together with a deep lagrangian cut, very hard problems like QAP (inherently non full dimensional) would become a challenging test to verify this approach. However solving (MKP) for itself would require a full hybridation of both global cuts and nested logic cuts (whose efficiency has already been mentioned).
References


Fig. 2. Multi knapsack relaxations and cuts.